ТЕКТРАН

СИСТЕМА ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

МАШИНОСТРОЕНИЕ

ТЕХТРАН –
СИСТЕМА ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

ЛЕНИНГРАД
«МАШИНОСТРОЕНИЕ»
ЛЕНИНГРАДСКОЕ ОТДЕЛЕНИЕ
1987

ББК 32.973
Т38
УДК 658.512.4.012.011.56

**А. А. Лиферов, О. Ю. Батунер, М. Ю. Блюдзе,
Н. А. Бурденков, С. Г. Королев**

Рецензент канд. техн. наук **А. М. Покровский**

Т 38 **Техтран** — система программирования оборудования с ЧПУ/ А. А. Лиферов, О. Ю. Батунер, М. Ю. Блюдзе и др.— Л.: Машиностроение, Ленингр. отд-ние. 1987.—109 с.: ил.

(В обл.): 55 к.

В книге на примере широко используемой в промышленности системы Техтран рассмотрены вопросы автоматизированного программирования оборудования с числовым программным управлением (ЧПУ). Приведены классификация систем автоматизированного программирования (САП) и функциональная структура системы Техтран. Большое внимание уделено описанию проблемно-ориентированного языка программирования Техтран, являющегося составной частью одноименной системы. Рассмотрены особенности технологической подготовки производства в условиях автоматизированной подготовки управляющих программ (УП) и приведены примеры программирования для различных типов оборудования.

Книга предназначена для инженерно-технических работников, занимающихся разработкой управляющих программ для оборудования с ЧПУ.

Т $\frac{2405000000-173}{038(01)-87}$ 173—87

**ББК 32.973
6Ф7.3**

© Издательство «Машиностроение», 1987

ПРЕДИСЛОВИЕ

Внедрение оборудования с числовым программным управлением (ЧПУ), средств вычислительной техники, а также создаваемых на их основе гибких производственных систем (ГПС) определяется современным развитием промышленного производства, которое характеризуют частая смена продукции, короткий жизненный цикл изделий, а также возрастающие сложность и точность обработки. В этих условиях одной из важнейших задач технологической подготовки производства является разработка управляющих программ (УП) для станков с ЧПУ.

Разработка УП — сложный и трудоемкий процесс, во многом определяющий эффективность использования оборудования с ЧПУ и качество обрабатываемых деталей, поэтому сейчас большое внимание обращается на разработку и совершенствование систем и методов автоматизированной подготовки УП. Особое внимание проблеме автоматизированной подготовки управляющих программ уделяется в составе комплексной системы САПР—АСТПП—ГПС. Управляющая программа при этом является результатом сквозного цикла обработки информации от чертежа детали до программы ее изготовления на станках с ЧПУ. Система автоматизированного программирования (САП) оборудования с ЧПУ — составная часть такой комплексной системы. Несмотря на большое число работ в области автоматизации программирования оборудования с ЧПУ в нашей стране и за рубежом, они не получили полного и систематизированного изложения в литературе. Предлагаемая книга содержит изложение основных вопросов, связанных с автоматизированной подготовкой УП для станков с ЧПУ на базе САП Техтран.

В гл. 1 дана классификация САП по различным критериям, позволяющая получить представление о направлениях развития автоматизации программирования, и приведена функциональная структура САП Техтран. Гл. 2 посвящена описанию проблемно-ориентированного языка программирования оборудования с ЧПУ Техтран. Описание приведено в полном объеме и может быть использовано на практике при разработке УП. Гл. 3 посвящена операторам постпроцессора, задающим необходимые технологические данные для выполнения УП на станке. В гл. 4 изложены особенности технологической подготовки производства в условиях использования САП, приведены примеры различных видов обработки и описание способов контроля УП.

В качестве основной задачи авторы ставили перед собой полное и систематизированное описание языка программирования оборудования с ЧПУ Техтран, изложенного в гл. 2. Гл. 1, 3, 4, включены для более глубокого восприятия основного материала.

Все замечания и пожелания авторы просят направлять по адресу: 191065, Ленинград, ул. Дзержинского, 10, ЛО издательства «Машиностроение».

Глава 1

АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

1.1. Системы автоматизации программирования

Роль систем автоматизации программирования в процессе подготовки управляющих программ для станков с ЧПУ можно сейчас сравнить с той ролью, которую сыграли компиляторы языков высокого уровня в процессе разработки программ на ЭВМ. Появление САП является логическим следствием развития вычислительной техники, программирования, станкостроения, а также обусловлено возрастающей сложностью деталей, и, следовательно, высокой трудоемкостью «ручного» программирования их обработки. За более чем двадцатилетний период, прошедший с тех пор, как начали эксплуатироваться САП, сменилось четыре поколения ЭВМ, появились микропроцессоры, разработаны и широко используются многооперационные станки типа «обрабатывающий центр» и роботы. Это естественным образом повлияло на методы подготовки управляющих программ и эволюцию развития САП.

Программирование оборудования с ЧПУ появилось совсем недавно. Результатом программирования является управляющая программа, которая представляет собой совокупность команд на языке программирования, соответствующая заданному алгоритму функционирования станка по обработке конкретной заготовки (см. ГОСТ 20523—80).

В книге будет идти речь об автоматизированном программировании, хотя последние достижения вычислительной техники и станкостроения в некоторой степени стирают грань между «ручным» и автоматизированным программированием. Рассмотрим программирование как составную часть процесса технологической подготовки производства.

«Ручное» программирование в традиционном понимании состоит в следующем. По заданному операционному технологическому процессу технолог рассчитывает траекторию движения инструмента, назначает необходимые технологические команды в соответствии с режимом обработки (например, подачу, охлаждение, смену инструмента и т. д.) и преобразует эти данные во внутреннее представление системы ЧПУ [4, 15, 17].

При автоматизированном программировании в идеальном случае все задачи ручного программирования должны решаться на ЭВМ, включая расчет режимов обработки. САП являются эффективным и надежным инструментом для их решения.

САП — это комплекс технических, программных, языковых, информационных средств, осуществляющих преобразование данных чертежа детали в коды устройства управления оборудованием с ЧПУ. Как правило, САП организованы по классической структуре: входной язык, процессор, промежуточный язык, постпроцессор. В зависимости от реализации возможны отклонения от этой структуры.

Входной язык САП — проблемно-ориентированный язык, предназначенный для описания исходных данных о детали и технологическом процессе ее обработки на оборудовании с ЧПУ.

Процессор САП — программное изделие, предназначенное для решения общих геометрических и технологических задач, а также задач управления процессом обработки данных на ЭВМ, реализация которого возможна в виде специализированного программного устройства.

Постпроцессор САП — программное изделие, предназначенное для адаптации управляющей программы к конкретному оборудованию с ЧПУ, реализация которого возможна в виде специализированного программного устройства.

Промежуточный язык процессор — постпроцессор — внутренний проблемно-ориентированный язык системы САП, служащий для представления данных, передаваемых

от процессора к постпроцессору. (Все определения даны по ГОСТ 23501.603—84).

В литературе промежуточный язык процессор-постпроцессор встречается под названием CLDATA (*Cutter Location Data* — данные о перемещении инструмента).

Структура САП приведена на рис. 1.1. Обычно в состав САП входит библиотека постпроцессоров, каждый из которой предназначен для определенной модели системы ЧПУ или модели станка. Разрабатываются также универсальные постпроцессоры, способные формировать УП для различных систем ЧПУ. Графические постпроцессоры предназначены для вывода траектории движения инструмента на графические отображающие устройства.

Рассмотрим классификацию САП по различным критериям (рис. 1.2) [9]. Первый критерий классификации САП — *по числу управляемых координат* — определяет

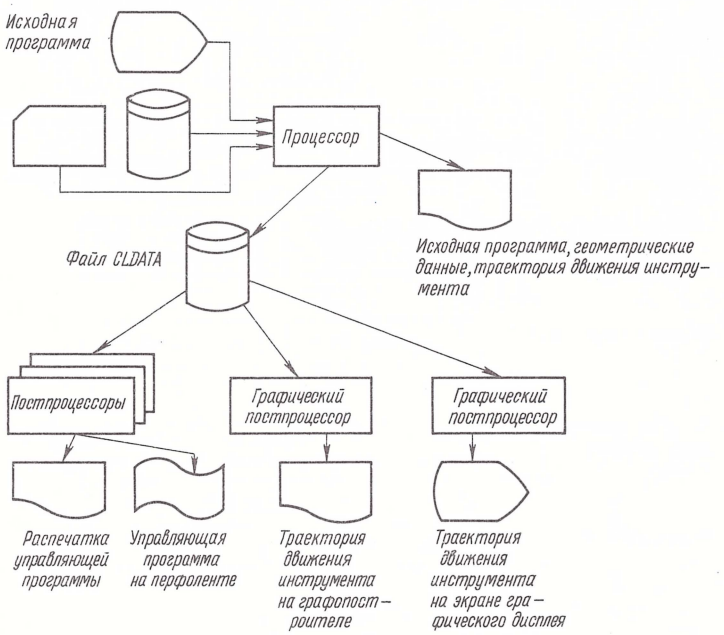


Рис. 1.1

функциональные возможности системы и класс обслуживаемого оборудования. Двухкоординатные САП обеспечивают автоматизированную подготовку УП для двухкоординатных станков (токарных, электроэрозионных, газорезательных и др.). Движение инструмента происходит в одной из координатных плоскостей. Движения по третьей координатной оси нет. 2,5-координатные САП обеспечивают подготовку УП для обработки произвольного плоского контура при условии, что поверхность детали является плоскостью, произвольно ориентированной в пространстве. Такие САП относятся к самому распространенному типу систем по двум причинам: во-первых, они охватывают подготовку УП для широкого класса станков (токарных, фрезерных, сверлильных и др.), во-вторых, трудоемкость разработки, сопровождения и эксплуатации такой САП не требует больших инженерных и технических ресурсов. К числу таких САП относятся ЕСПС-ТАУ, САП-СМ, Техтран, ADAPT (США), NELAPT (Великобритания), Autotech (ГДР), FAPT (Япония), IFAPT (Франция). В трехкоординатных САП в отличие от 2,5-координатных поверхность обрабатываемой детали может быть произвольной поверхностью второго порядка, что позволяет расширить область их применения. Типичными представителями таких САП являются MODAPT (Италия), PROMO (Франция). Многокоординатные САП в дополнение к обычным трехкоординатным перемещениям обеспечивают угловые перемещения вокруг одной из координатных осей. Реализация угловых перемещений производится постпроцес-

сором и зависит от того, каким органом станка производится поворот — шпиндельной головкой или столом. Наиболее значительной из семейства многокоординатных САП является разработанная в 1961 г. в США система АРТ (*Automatically Programmed Tools*).

Следующий критерий классификации — *по уровню принятия решений* — определяет уровень автоматизации решаемых в САП технологических задач. К технологическим задачам относятся реализация типовых технологических циклов точения, сверления, нарезания резьбы, фрезерования кругового и прямоугольного (внутреннего и наружного), фрезерования пазов и карманов, разбиения припуска на проходы, расчет режимов резания и др.

В настоящее время трудно назвать САП, в которой не решалась бы одна из названных задач. Поэтому основной принцип классификации по этому критерию

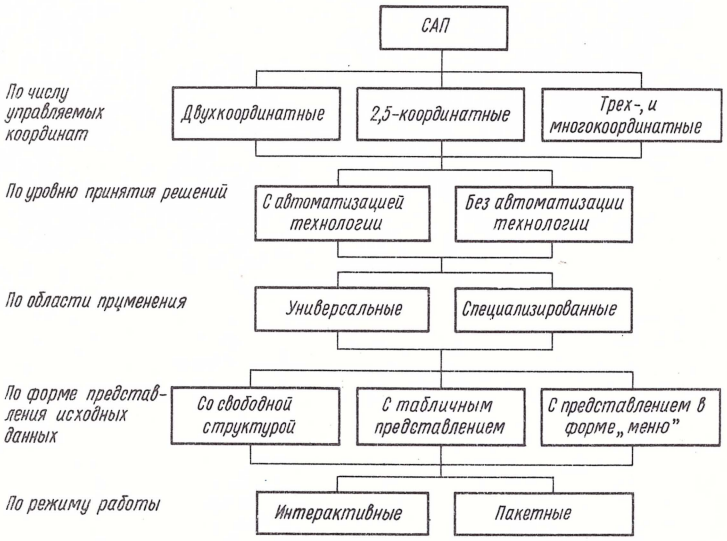


Рис. 1.2

состоит в том, является автоматизация технологических задач свойством, заложенным в концепцию САП при ее разработке, или такая автоматизация реализована на основе библиотеки типовых технологических процедур. В большинстве САП (например, САП-СМ, АРТ, FAPT, IFAPT, MODAPT) автоматизация технологических задач решается при помощи технологических процедур. Среди САП с встроенной автоматизацией технологических задач можно назвать ЕСПС-ТАУ, Техтран-Т, Autotech, EXAPT (ФРГ), FANUC-3T и MAZATROL (Япония).

В зависимости от *области применения* САП можно разделить на универсальные и специализированные. Универсальные САП — это системы широкого назначения, с развитым входным языком, позволяющим разрабатывать программы, имеющие сложную логическую структуру, описывать обработку произвольных контуров, производить математические вычисления, а также создавать типовые технологические процедуры для различных классов оборудования. АРТ является типичной САП этого класса. Она представляет собой обобщенную универсальную САП, предназначенную для программирования обработки сложных поверхностей, с управлением по трем и пяти осям [5]. АРТ является прототипом семейства САП, разработанных в различных странах, в том числе в СССР. В частности, за основу языка Техтран было взято подмножество языка АРТ. АРТ позволяет программировать обработку поверхностей типа конусов, цилиндров, поверхностей второго порядка (эллипсоид, эллиптический конус, гиперболический параболоид и др.), линейчатых и каркасных поверхностей, сфер. АРТ является многофункциональной системой, но основными областями

ее эффективного применения являются авиа-, судо- и автомобилестроение, где наиболее часто встречается обработка сложных поверхностей (например, формообразующих элементов штампов, гребных винтов, элементов конструкций самолета и др.). Несмотря на высокую производительность и большие возможности, система APT не нашла широкого применения вследствие сложности в изучении и эксплуатации [5].

Специализированные САП классифицируются по видам обработки (токарной, фрезерной, сверлильно-расточной и др.). Последние годы развитие САП идет по пути создания специализированных систем с высоким уровнем автоматизации решения технологических задач.

В ФРГ разработаны семейства САП EXAPT, состоящие из технологически ориентированных подсистем, в основу которых были положены концепции APT [18].

Типичным представителем специализированных САП является EXAPT (ФРГ). EXAPT состоит из трех подсистем. EXAPT-M01 (Basic — EXAPT) — универсальная САП — ядро системы; EXAPT-M02 — специализированная, технологически ориентированная САП для токарных станков, EXAPT-M03 — для сверлильных, фрезерных и станков типа обрабатывающий центр. Программирование в EXAPT-M02 состоит из следующих этапов: описания геометрии детали, описания технологических переходов, описания процесса обработки. При этом автоматизировано решение следующих задач: разбиение на проходы, расчет режимов резания, построение траектории движения инструмента при черновой и чистовой обработке, контроль на наличие столкновений.

В EXAPT-M03 автоматизированы задачи выбора инструмента, расчета ходов инструмента при позиционной и контурной обработке, расчет режимов резания, контроль на наличие столкновений.

Система EXAPT является наиболее типичным представителем семейства специализированных САП. Среди множества других САП можно назвать ориентированную на токарную обработку TAUPROG (Венгрия) и отечественные ЕСПС-ТАУ для сверлильных, токарных и фрезерных работ и Техтран-Т для токарных работ, в которых также решены перечисленные технологические задачи.

В настоящее время выпускаются системы ЧПУ типа CNC со встроенными САП — программаторы. Программаторы — это специализированные САП, предназначенные для оперативной подготовки УП непосредственно на станке. Это позволяет сократить простои станка за счет совмещения обработки детали с разработкой, отладкой и редактированием другой УП. Наибольший интерес представляют системы BOSCH-PEG (ФРГ) и FANUC-3T (Япония) — для токарных станков, FANUC 6M-MODEL E и MAZATROL-M (Япония) — для обрабатывающих центров [14, 16].

По *форме представления исходных данных* можно выделить САП со свободной структурой представления, с табличной формой и представлением в форме «меню».

Большинство известных САП имеют свободную форму представления исходных данных. Программа на языке таких САП представляет собой множество операторов, состоящих из элементов языка (служебных слов, идентификаторов, специальных символов и т. д.) и строящихся по заданному синтаксису (например, входные языки многих из названных САП соответствуют синтаксису языка APT). В дальнейшем для удобства будем отождествлять названия САП и ее входного языка.

При табличной форме представления технолог заполняет специальные бланки в форме таблиц. Каждая графа таблицы соответствует определенному виду исходных данных, например, геометрии детали, технологическим командам, управляющей информации и т. д. Имеются также САП, сочетающие обе формы представления (САДКО, АППР-АРМ).

Представления в форме «меню» — это свойство интерактивных САП. С экрана дисплея в определенном формате запрашивается требуемая информация. Технологу остается только заполнить соответствующие графы, высвечиваемые на экране дисплея, или ответить на задаваемые системой вопросы. Наиболее перспективными являются САП, сочетающие представление исходных данных со свободной структурой и в форме «меню», например, в системе ТС-APT [20].

Последний критерий классификации САП — по *режиму работы*. В настоящее время существуют два основных режима работы САП: пакетный и интерактивный, которые находятся в прямой зависимости от типа ЭВМ и операционной системы, для которых реализована данная САП. Первые САП работали в пакетном режиме. Программа, подготовленная технологом на входном языке САП и записанная на магнитный носитель, перфоленту или перфокарты, полностью вводилась в ЭВМ и преобразовывалась в УП для станка. В случае ошибок в исходной программе эта процедура повторялась.

Разработка интерактивных САП является следствием развития вычислительной техники — появлением алфавитных и графических дисплеев, а также мини- и микро-ЭВМ. Исходные данные в интерактивных САП задаются в форме «меню» или в режиме «вопрос — ответ». Существуют интерактивные САП и со свободной формой представления (например, FAPT, где последовательно после ввода каждого оператора можно получить на экране дисплеев или печатающем устройстве результат его выполнения). Значительное преимущество таких САП перед пакетными состоит в сокращении сроков обучения технологов-программистов и технологов-операторов методам программирования за счет диалога и возможности повторения расчета УП с любой исходной позиции. Недостатком их является отсутствие универсальности в подходе к разработке УП вследствие того, что диалог ограничивает разработчика в выборе средств для решения задачи. Интерактивные САП эффективны при эксплуатации в производствах с невысокой сложностью и высоким уровнем унификации изготавливаемых деталей или их элементов.

Представленная классификация САП является попыткой систематизировать накопленный опыт и направления развития данной области инженерной деятельности за последние годы и носит условный характер. В последующие годы параллельно с развитием диалоговых, оперативных систем подготовки УП непосредственно у станка дальнейшее развитие получат системы программирования, работающие в комплексных системах автоматизированного проектирования и производства (CAD/CAM — *Computer Aided. Design/Computer Aided Manufacturing*). Комплексная система состоит из ряда подсистем, с помощью которых производится формирование цифровой модели детали, вычерчивание рабочих чертежей и эскизов на графопостроителях, проектирование технологического процесса и формирование управляющей программы [19]. Объединение этих процессов в единый непрерывный процесс переработки информации и управления технологическими процессами и оборудованием дает ряд преимуществ перед описанными ранее системами и методами. Преимущества эти следующие: сокращение затрат и времени на подготовку УП; повышение качества УП; повышение эффективности использования оборудования с ЧПУ за счет быстрой загрузки программ в память системы ЧПУ и отсутствие брака в УП; высвобождение технологов-программистов, занимающихся разработкой УП; сокращение сроков подготовки производства новых изделий.

В частности, имеется опыт создания комплексной системы проектирования технологических процессов и подготовки УП для номенклатуры деталей тел вращения, функционирующей в условиях гибкой производственной системы «Группроцесс-Техтран». Подробное описание перечисленных САП приведено в работах [1, 3, 5, 7, 10, 12, 13, 14, 16, 20, 22, 23].

1.2. Структура и функциональные возможности САП Техтран

В основу организации САП Техтран был положен традиционный подход. Система состоит из входного языка, процессора, промежуточного языка и постпроцессоров. При разработке системы преследовалось несколько целей. Во-первых, САП должна быть простой в обслуживании, т. е. вмешательство оператора должно быть сведено к минимуму. Во-вторых, САП должна быть мобильной (это позволяет минимизировать затраты при адаптации САП для различных классов ЭВМ и операционных систем). В-третьих, САП должна допускать расширение выполняемых функций.

Необходимую гибкость САП Техтран приобрела благодаря модульной структуре. Для уменьшения объема оперативной памяти процессор и постпроцессор выполняют обработку данных за несколько проходов: процессор построен по трехпроходной схеме, постпроцессоры — по двухпроходной схеме. Модульная структура дает возможность вводить новые функции без существенной корректировки ранее написанных программ. Кроме того, такая структура позволяет организовать перекрытия и уменьшить тем самым объем оперативной памяти.

Требование мобильности программного обеспечения предопределило выбор языка программирования. Поскольку ко времени начала разработки (1975 г.) и вплоть

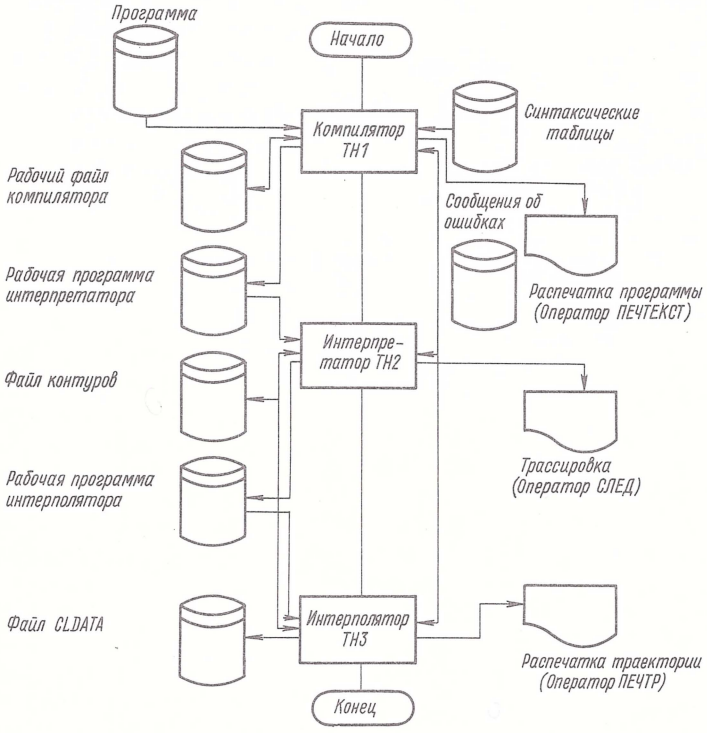


Рис. 1.3

до настоящего времени единственным языком программирования, компиляторы которого входят в состав всех операционных систем и обеспечивают необходимое качество объектного кода, был и остается Фортран, то именно ему и было отдано предпочтение. Около 95 % программ, составляющих САП Техтран, написано на Фортране, остальные — на Ассемблере. В настоящее время синхронные версии системы эксплуатируются на ЭВМ типа ЕС с операционной системой ОС ЕС и СМ с операционной системой ОС РВ.

Процессор осуществляет ввод и трансляцию исходной программы на входном языке системы, выполняет заданные геометрические построения и рассчитывает траекторию движения центра инструмента. Результаты расчета выводятся в файл CLDATA. Структура процессора системы Техтран приведена на рис. 1.3. Процессор организован в виде трех последовательно выполняемых модулей: компилятора, интерпретатора и интерполятора. Каждый модуль может рассматриваться как законченная программа. Обмен данными между модулями производится только через промежуточные файлы на диске.

Компилятор вводит программу на Техтране, предварительно записанную на диск, и выполняет ее синтаксический контроль. На этапе компиляции проверяется, правильно ли заданы имена и служебные слова в программе, соблюдены ли требуемые

форматы. Если часть текста оформлена в виде отдельного файла, то компилятор вставляет этот текст в исходную программу. Макросредства дают программисту возможность описывать типовые элементы в параметрическом виде и хранить их на магнитном диске. В процессе макрогенерации компилятор выполняет вставку тела макроса в исходную программу и заменяет формальные параметры фактическими значениями. Если обнаружена ошибка, то компилятор отмечает ее место в строке и печатает диагностическое сообщение. Затем он переходит к следующему оператору, чтобы за один прогон выявить максимальное число ошибок. Файл синтаксических таблиц содержит описание форматов операторов Техтрана и служит для управления работой компилятора. Тексты сообщений об ошибках также хранятся на магнитном диске. Компилятор формирует на диске рабочую программу интерпретатора, в которой арифметические выражения преобразованы в польскую инверсную запись, геометрические определения преобразованы во внутренний формат и заданные в них модификаторы упорядочены. В процессе макрогенерации используется рабочий файл на диске.

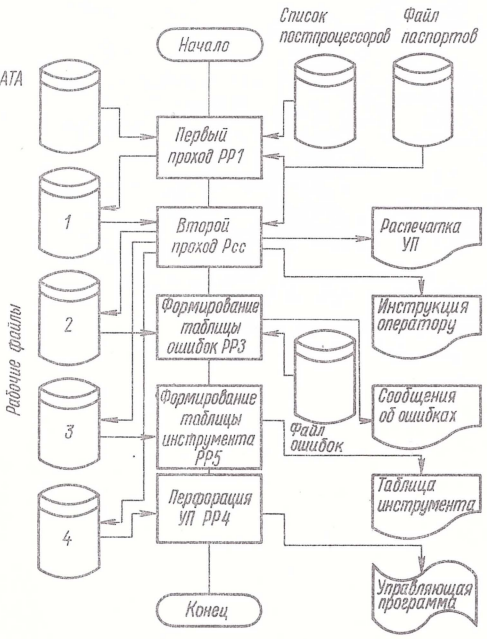


Рис. 1.4

Интерпретатор производит расчет канонических форм геометрических объектов. Другой функцией интерпретатора является вычисление значений арифметических и логических выражений и стандартных функций. Если в программе были использованы операторы цикла и операторы передачи управления, то интерпретатор выполняет эти операторы и тем самым приводит исходную программу к линейной. Рабочая программа интерполятора, формируемая интерпретатором, будет содержать только таблицу геометрических объектов, операторы движения и операторы постпроцессора. Интерпретатор может выполнять трассировку, т. е. распечатывать значения переменных и переходов в программе. Если в исходной программе были определены контуры, то их описания во внутреннем формате выводятся в специальный файл. При обнаружени ошибки в программе интерпретатор печатает диагностическое сообщение и продолжает обработку до заданного числа ошибок.

Интерполятор обрабатывает операторы движения и рассчитывает эквидистантную траекторию с учетом диаметра рабочего инструмента. Если интерполятор обнаруживает ошибку, то он выдает диагностическое сообщение и продолжает обработку от следующего оператора поточечного движения до заданного числа ошибок. В процессе вычислений интерполятор использует рабочую программу и файл контуров, подготовленные интерпретатором, а также файл сообщений об ошибках. Операторы постпроцессора не обрабатываются и передаются в выходной файл. Выходные данные интерполятора формируются в стандартном формате CLDATA.

Процессор рассчитан на обработку группы исходных программ за один проход. Каждый модуль процессора за один вызов обрабатывает всю группу. Программы, в которых на стадии компиляции или интерпретации обнаружены ошибки, удаляются из соответствующего выходного файла и не поступают на дальнейшую обработку. Программы с ошибками в операторах движения выводятся в файл CLDATA, но место ошибки отмечается специальной записью.

Постпроцессоры выполняют следующие функции: чтение файла CLDATA, подготовленного процессором, и синтаксический контроль операторов постпроцессора; перевод геометрических данных в систему координат станка; проверку длины перемещений, допустимости подач и оборотов; расчет кодов подач, кодов оборотов шпинделя, коррекции; учет динамических характеристик оборудования; формирование и вывод на печать и перфоленту управляющей программы и другие функции.

САП Техтран оснащена рядом постпроцессоров, каждый из которых формирует управляющие программы для определенной системы управления. Настройка постпроцессора на конкретную модель станка производится с помощью паспорта. Структура постпроцессора приведена на рис 1.4.

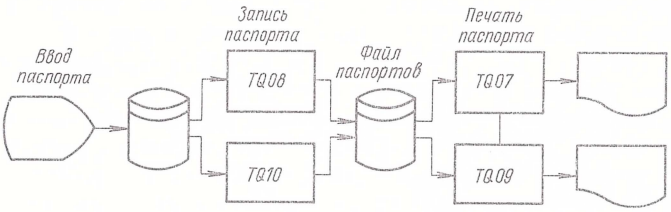


Рис. 1.5

Постпроцессоры построены по двухпроходной схеме. Программа первого прохода *PP1* читает записи CLDATA, сглаживает траекторию, проверяет, принадлежат ли перемещения рабочей зоне станка, рассчитывает вектор коррекции и формирует в рабочем файле *1* исходные данные для второго прохода. Программа второго прохода *Pcc* [1] выполняет синтаксический контроль операторов постпроцессора, учитывает динамические особенности оборудования, формирует на диске образ управляющей программы и другие рабочие файлы, распечатывает управляющую программу и инструкцию оператору станка. Постпроцессор формирует три рабочих файла, каждый из которых обрабатывается специальной программой:

рабочий файл *2* содержит коды сообщений об ошибках; он обрабатывается программой *PP3*, которая печатает таблицу сообщений об ошибках и таблицу недопустимых для данного станка операторов;

рабочий файл *3* содержит таблицу инструментов; он обрабатывается программой PP5, которая печатает таблицу инструмента и таблицу времени работы каждого инструмента;

рабочий файл *4* содержит управляющую программу во внутреннем формате (во внутреннем коде ЭВМ); он обрабатывается программой PP4, которая выполняет перекодировку в код системы ЧПУ и перфорирует ее.

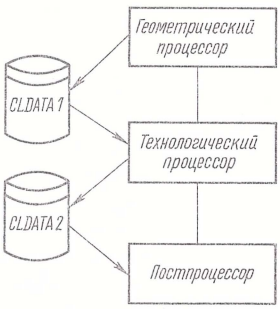


Рис. 1.6

Пользователям САП Техтран предоставлена возможность заполнять паспорта для имеющегося у них оборудования с ЧПУ при условии, что в составе библиотеки постпроцессоров имеется модуль для требуемой системы управления. Для работы с паспортами предназначена подсистема Вопросник, состоящая из четырех программ. Структура Вопросника показана на рис. 1.5. Паспорт состоит из двух частей — характеристик оборудования и таблиц. Каждая часть паспорта обрабатывается двумя программами: одна предназначена для записи паспорта (*TQ08* и *TQ10*), другая — для его распечатки (*TQ07* и *TQ09*). Вопросник читает исходный текст паспорта и проверяет форматы и значения данных. После перекодировки паспорт в двоичном формате записывается в файл паспортов, который используется при прогоне постпроцессоров.

[1] Идентификация постпроцессоров осуществляется с помощью двухсимвольного кода постпроцессора сс. Например, постпроцессор для системы управления Н-33 называется М1, а для системы Н-22 — L1.

Графический контроль в САП Техтран решен с помощью постпроцессоров, которые либо формируют перфоленту для графопостроителей, работающих в автономном режиме, либо непосредственно выводят на экран или бумагу траекторию перемещения центра режущего инструмента.

Технологический процессор является расширением геометрического процессора Техтран в части программирования технологических переходов, а входной язык технологического процессора — расширением базового языка, который дополняется технологическими операторами. Выделение геометрического процессора в автономный модуль позволило определить пути такого расширения системы Техтран, при котором ядро остается неизменным. Расширение функций системы осуществляется включением технологических процессоров специального назначения. Положение технологического процессора в системе Техтран показано на рис. 1.6.

Процессор формирует файл *CLDATA1*, в котором содержатся геометрические и технологические данные. Технологические операторы обрабатываются процессором так же, как и операторы постпроцессора: процессор заменяет служебные слова их внутренними кодами и вычисляет арифметические выражения. Расширенный файл *CLDATA1* обрабатывается технологическим процессором, на выходе которого формируется файл *CLDATA2*, содержащий траекторию движения инструмента и операторы постпроцессора.

Принцип расширения функций системы Техтран реализован в системе Техтран-Т — технологическом процессоре для программирования токарных переходов.

Глава 2

ВХОДНОЙ ЯЗЫК ТЕХТРАН

2.1. Основные понятия

Программа. Структура программы. Программа на Техтране представляет собой последовательность операторов и комментариев, описывающих алгоритм обработки деталей. Программа имеет следующую структуру:

ДЕТАЛЬ *литерал*
текст программы
КОНЕЦ

Литерал (см. п. 2.2) задает имя программы. В программу могут быть вставлены фрагменты исходного текста и тексты макросов. Программе на входном языке Техтран может соответствовать одна или несколько управляющих программ. Исходный текст программы состоит из последовательности строк. Максимальное число символов в строке равно 72. В одной строке может быть записан один или несколько операторов, между которыми должен стоять символ «точка с запятой» (;). Все символы текущего оператора должны размещаться до 72-й позиции. В противном случае часть оператора может быть перенесена на следующую строку. Для этого, после прерванной части оператора (до 72-й позиции) необходимо записать символ продолжения «процент» (%). Для совместимости с предыдущими версиями языка разрешается в качестве символа продолжения использовать знак «коммерческое И» с пробелом (&␣). Число строк продолжения не ограничивается.

Текст оператора или комментария может начинаться с любой позиции. Число пробелов между операндами не ограничено. В качестве разделителя операндов вместо запятой можно использовать пробел (␣). Для удобства чтения программ и ускорения их отладки рекомендуется располагать операторы и комментарии таким образом, чтобы текст программы являлся документом, по которому можно понять

структуру и назначение программы. Желательно, чтобы имена меток и переменных, используемых в программе, отражали их назначение.

Для выделения функциональных частей программы рекомендуется использовать строки, состоящие из пробелов, знаков комментария, а также отступов.

Символические имена. Для идентификации в программе переменных величин (геометрических, вещественных, логических, меток операторов, имен макросов) используют символические имена. Символическое имя — это последовательность букв и цифр, начинающаяся с буквы. Длина его не должна превышать восьми символов. Пробелы внутри имени не разрешены. Символическое имя должно быть уникальным в программе и описывать один объект или класс объектов, например:

АБ — верно
КР15 — верно
К152АБ — верно
К 1 — неверно
К123455678910 — неверно
123К22 — неверно

Операторы, метки, комментарии. Оператор — основная функциональная единица языка. Он предназначен как для описания операций, которые предстоит выполнить, так и для установления свойств данных, идентифицируемых с помощью символических имен. Любой оператор может быть помечен меткой, для того чтобы на него можно было сослаться в других операторах данной программы.

Метка является символическим именем. Она должна предшествовать помечаемому оператору и может находиться либо в одной строке с оператором, либо на отдельной строке. После метки всегда должен быть служебный символ «двоеточие» (:). Метка используется в операторах НАМЕТКУ.

Для пояснения действий, выполняемых в программе, используют комментарии. Комментарий не обрабатывается процессором, а только выводится на печать при распечатке исходного текста. Комментарий может находиться как в одной строке с операторами исходной программы, так и занимать целую строку. Для выделения текста комментария нужно поставить перед ним служебный символ «номер» (≠).

2.2. Структура языка

Алфавит языка. Для записи программы используют следующие символы: буквы русского алфавита А—Я; буквы латинского алфавита A—Z; цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; символы: «␣» — пробел; «,» — запятая; «.» — точка; «=» — равно; «+» — плюс; «—» — минус; «*» — звездочка; «/» — наклонная черта; «&» — коммерческое «И»; «;» — точка с запятой; «(» — левая скобка; «)» — правая скобка; «:» — двоеточие; «′» — апостроф; «≠» — номер; «%» — процент.

Синтаксис языка. В связи с тем, что в описании языка не приводится формального описания синтаксиса Техтрана, этот раздел содержит основные правила и обозначения, используемые для описания форматов его операторов. Каждый оператор входного языка формируется из операндов в соответствии с синтаксическими правилами языка. В описании языка приведена структура каждого оператора (формат).

При описании формата каждого оператора использованы следующие правила и условные обозначения:

прописные буквы используют для обозначения служебных слов;

строчные буквы используют для обозначения опорных элементов, числовых значений параметров или арифметических выражений;

необязательные служебные слова заключают в квадратные скобки [ ];

возможные варианты записи модификаторов, являющиеся альтернативными, заключают в фигурные скобки { } и записывают в столбец или разделяют символом «|»;

параметры, назначающие линейные и угловые размеры, задаются целыми и десятичными долями миллиметров и градусов.

Элементы языка. В языке разницы между целыми и вещественными числами не существует. *Число* представляется в виде последовательности цифр, среди которых может встретиться точка или буква Е. Точка отделяет целую часть числа от дробной, а буква Е — мантиссу числа от его порядка. Перед самим числом и перед его порядком может стоять знак «минус» (—). Пробелы и другие символы внутри числа недопустимы, например: 1.23 — верно; 2.15E5 — верно; —3E—6 — верно; 3 E2 — неверно; .E4 — неверно.

Служебные слова являются зарезервированными словами и составляют словарь языка Техтран. Они используются для написания операторов программы и состоят только из букв русского алфавита. Служебные слова делятся на главные, вспомогательные и модификаторы. Главное слово определяет тип оператора. Вспомогательное слово и модификатор служат для однозначного задания оператора. Служебное слово не может использоваться в качестве имени, переменной или метки. Список служебных слов приведен в Приложениях 1—4.

Переменная в Техтране — это символическое изображение величины, которая может принимать различные значения. Основными характеристиками переменной являются ее имя и тип.

В группу *служебных символов* входят следующие символы: «␣», «+», «—», «*», «/», «=», «;», «.», «,», «:», «'», «≠», «&», «( )», «%». Употребление служебных символов в программе зависит от контекста. Описание их функций производится по мере надобности в соответствующих разделах.

Литералом является заключенный в апострофы произвольный текст, не содержащий символа ≠. В литералах могут быть использованы любые символы (например, латинские буквы, !, >, < и т. п.). При необходимости использования апострофа его нужно записать два раза подряд, например, 'АВСДЕ'' РОН' определяет литерал, состоящий из символов АВСДЕ' РОН. В основном литералы используются для задания текстовых параметров макросов, но кроме этого имеются и некоторые другие применения, например, в операторах ДЕТАЛЬ, СТАНОК и др.

2.3. Типы данных

Возможности языка программирования в значительной степени определяются набором допустимых типов данных. Входной язык Техтран допускает применение следующих типов данных: вещественный, логический, вектор, прямая, точка, окружность, плоскость, матрица, контур. Тип каждой используемой в программе переменной фиксирован, и на протяжении всей программы каждая переменная обозначает данное только одного типа. Тип переменной описывается явно строкой вида:

тип имя$_1$, *имя*$_2$, *имя*$_3$, . . .

где *тип* есть: ВЕЩ, ЛОГИЧ, ТОЧКА, ПРЯМАЯ, ВЕКТОР, ОКРУЖН, ПЛОСК, МАТР, КОНТУР; *имя*$_1$, *имя*$_2$, *имя*$_3$, . . . — имена переменных.

Переопределение типа переменной в программе не разрешено. Если в программе встречается переменная, не описанная ранее, то имя переменной, начинающееся с символов ТЧ, считается неявно описанным, как точка; имя, начинающееся с ПР,— как прямая, с КР,— как окружность. Все остальные имена переменных определяются неявным образом, как вещественные числа.

Можно совмещать описание типа переменной и оператор присваивания, например, оператор ТОЧКА Т1 = 0, 1, 2 равносилен двум операторам: ТОЧКА Т1; Т1 = 0, 1, 2.

2.4. Массивы

Данные одного типа могут объединяться в массивы. Массивы могут быть одномерными и двумерными. Каждый используемый в программе массив должен быть описан с указанием типа, числа индексов (размерности массива) и максимального значения каждого из индексов. Минимальным значением индекса является единица.

Оператор ВЕЩ А (16) описывает одномерный массив А, состоящий из 16 вещественных чисел. Оператор ВЕЩ В (5, 6) описывает двумерный массив, состоящий из 30 вещественных чисел. Для использования элементов массива в вычислениях следует указать имя массива, а за ним в круглых скобках — значение индексов. Индекс не может быть опущен. В частности, одно имя не представляет первый элемент массива.

В качестве индекса может использоваться любое арифметическое выражение. Число индексов и их значения должны соответствовать описанию массива. Число элементов в массиве не ограничено. Например, если значение переменной И=2, то запись В (3, И+3) именует элемент массива В (3,5), а запись вида В (3) также, как и запись В (1,6+И), вызывает ошибку трансляции.

В качестве элементов массива не могут использоваться объекты типов МАТР и КОНТУР.

2.5. Выражения

Арифметические выражения. Вместо чисел в программе могут использоваться арифметические выражения. Простейшими арифметическими выражениями могут быть число; вещественная переменная; элемент вещественного массива; обращение к стандартной функции; выражение, заключенное в скобки.

Более сложные арифметические выражения могут быть образованы из простейших при помощи следующих знаков арифметических операций: «+» — сложение; «—» — вычитание; «*» — умножение; «/» — деление; «**» — возведение в степень.

Операции выполняются слева направо в соответствии с приведенными ниже приоритетами операций: вычисление функций — 1; возведение в степень — 2; умножение и деление — 3; сложение и вычитание — 4.

Приоритет используется для того, чтобы определить, какая из двух последовательных операций выполняется первой. Для изменения порядка вычислений следует применять круглые скобки.

Примечание. При возведении в степень следует помнить, что отрицательное число допускается возводить только в целую степень.

Например, выражение $\frac{6a^2+2b\,(c^3-d^3)}{(a^2+b^2)\sin f}$ записывается на входном языке Техтран так:

(6*А**2+2*В* (С**3—Д**3))/((АА**2+В**2)*СИН(Ф)).

Логические выражения. Язык Техтран допускает использование логических выражений в условных операторах (ЕСЛИ), операторах цикла (ПОВТОР), операторах логического присваивания. Значением логического выражения является ЛОЖЬ или ИСТИНА.

Простейшее логическое выражение состоит из логического первичного. Логическим первичным может быть: логическая константа ЛОЖЬ и ИСТИНА; логическая переменная; элемент логического массива; оператор корректности; отношение; логическое выражение, заключенное в скобки.

Отношения образуются путем объединения двух арифметических выражений с помощью операций отношения. Шесть операций отношения изображаются следующим образом:

БЛШ или > — больше (>)
БРВ или > = — больше или равно (≥)
МНШ или < — меньше (<)
МРВ или < = — меньше или равно (≤)
РВН или = — равно (=)
НРВ или < > или < > — не равно (≠)

Например, выражение (К+1) РВН 5 принимает значение ИСТИНА при К=4, в остальных случаях принимает значение ложь.

Логические выражения образуются из логических первичных путем использования логических операций. В языке Техтран существуют три логические операции: НЕ — отрицание; ЛИ — логическое умножение; ИЛИ — логическое сложение.

Вычисление логического выражения производится слева направо в соответствии с приведенными ниже приоритетами операций: вычисление функций — 1; возведение в степень — 2; умножение и деление — 3; сложение и вычитание — 4; операции отношения БЛШ, БРВ, МНШ, МРВ, РВН, НРВ — 5; НЕ — 6; ЛИ — 7; ИЛИ — 8.

Для изменения порядка вычислений можно использовать круглые скобки. Логическое выражение, к которому относится операция НЕ, должно заключаться в скобки, если оно содержит не менее двух операндов.

2.6. Стандартные функции

Язык Техтран содержит ряд стандартных математических функций, часто используемых при вычислениях. Вызов математической функции производится указанием в арифметическом выражении имени функции и ее аргументов (параметров). Результат, вычисленный функцией, замещает в выражении обращение к ней. Стандартные функции в языке могут быть вещественного (табл. 2.1) и логического (табл. 2.2) типов.

Таблица 2.1

Имя	Функция	Аргумент	Результат
СИН	Вычисление синуса	*число*	Синус
КОС	» косинуса	»	Косинус
ТАН	» тангенса	»	Тангенс
АСИН	» арксинуса	»	Арксинус
АКОС	» арккосинуса	»	Арккосинус
АТАН	» арктангенса	»	Арктангенс
КВКОР	Извлечение квадратного корня	»	Квадратный корень числа
ЕХР	Вычисление экспоненты e^x	*число* (x)	e^x
ЛОГ	Вычисление натурального логарифма	*число*	Натуральный логарифм числа
АБС	Вычисление абсолютного значения величины	»	Абсолютное значение величины
МОДВ	Вычисление модуля вектора	*вектор*	Модуль вектора
ЦЕЛ	Выделение целой части переменной	*число* или *имя*	Целая часть числа или переменной с заданным именем
УГОЛВ	Вычисление угла между двумя векторами	$вектор_1$, $вектор_2$	Угол между двумя векторами
РАСТ	Вычисление расстояния между двумя точками	$точка_1$, $точка_2$	Расстояние между двумя точками
КООРД	Извлечение составляющей канонической формы	*число* (a), *имя*	Составляющая под номером «а» канонической формы геометрического объекта с заданным именем
СКАЛПР	Вычисление скалярного произведения	$вектор_1$, $вектор_2$	Скалярное произведение векторов
ДИСТ	Вычисление расстояния от точки до прямой	*точка, прямая*	Расстояние между точкой и прямой
УГОЛПР	Вычисление угла между прямыми	$прямая_1$, $прямая_2$	Угол между прямыми
ЛОГ10	Вычисление десятичного логарифма числа	*число* (a)	$\lg a$

Примечания: 1. Аргументы заключаются в круглые скобки и перечисляются через запятую. 2. Вместо чисел и имен переменных в качестве аргументов могут использоваться арифметические выражения и вложенные определения. 3. Угол между векторами или прямыми отсчитывается от элемента, заданного в качестве первого аргумента. 4. Значения стандартных функций могут использоваться в качестве оператора.

Таблица 2.2

Имя	Функция	Аргумент	Результат
КОРР	Корректность	Вложенное геометрическое определение	ИСТИНА, если геометрический объект может быть построен; ЛОЖЬ — в противном случае
НЕКОРР	Некорректность	Вложенное геометрическое определение	ИСТИНА, если геометрический объект не может быть построен; ЛОЖЬ — в противном случае

Пример использования логической функции КОРР:

ЛОГИЧ А=КОРР ТОЧКА ТЧ1 (ПЕРЕСЕЧ, ПР1, КР1, УБ)

ЕСЛИ (НЕ А) ТЧ1=ТЧ0.

Переменной ТЧ1 присваивается значение одной из точек пересечения окружности и прямой, если они пересекаются; в противном случае переменной ТЧ1 присваивается значение переменной ТЧ0.

2.7. Оператор присваивания

Для присваивания значения переменным используется оператор присваивания. Он состоит из левой части, знака равенства и правой части. В результате выполнения оператора присваивания значение правой части присваивается переменной, находящейся в левой части (тип левой части должен допускать такое присваивание). Если в левой части стоит переменная вещественного (логического) типа, то и в правой части должно стоять арифметическое (логическое) выражение. Например: оператор X=X+1 — правильный оператор присваивания, в результате выполнения которого значение переменной X будет увеличено на 1.

2.8. Система координат

В системе Техтран все объекты и траектория движения рассматриваются и описываются в правосторонней прямоугольной системе координат (см. рис. 2.1). На рис. 2.2, *а—в* приведены примеры отсчета координат на плоскости (рис. 2.2, *а*), в пространстве (рис. 2.2, *б*) и угловых размеров (рис. 2.2, *в*). Определяемые прямые и окружности всегда находятся в горизонтальной плоскости *xOy*, поэтому описания геометрии и движения по контуру и поверхности детали не должны зависеть от системы координат конкретного станка. Рассчитанные процессором системы перемещения (координаты) приводятся в соответствие с фактическими направлениями и адресами координат станка при работе постпроцессора, формирующего управляющую программу.

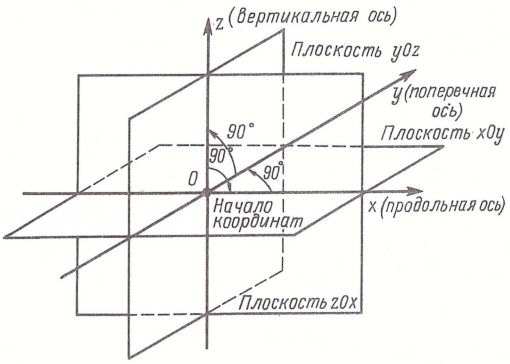


Рис. 2.1

На рис. 2.3, *а—г* приведены примеры перехода от системы координат детали к системе координат станка для токарных (рис. 2.3, *а*, *б*) и фрезерных (рис. 2.3, *в*, *г*) станков.

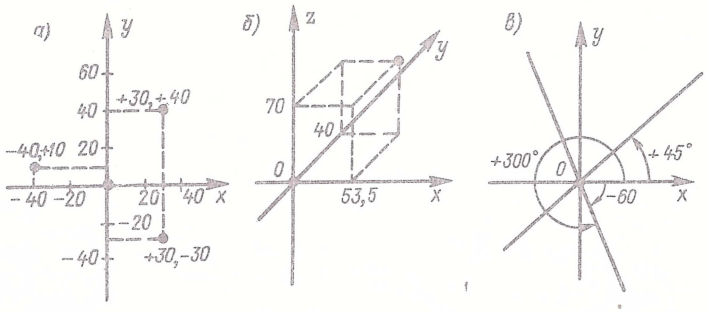


Рис. 2.2

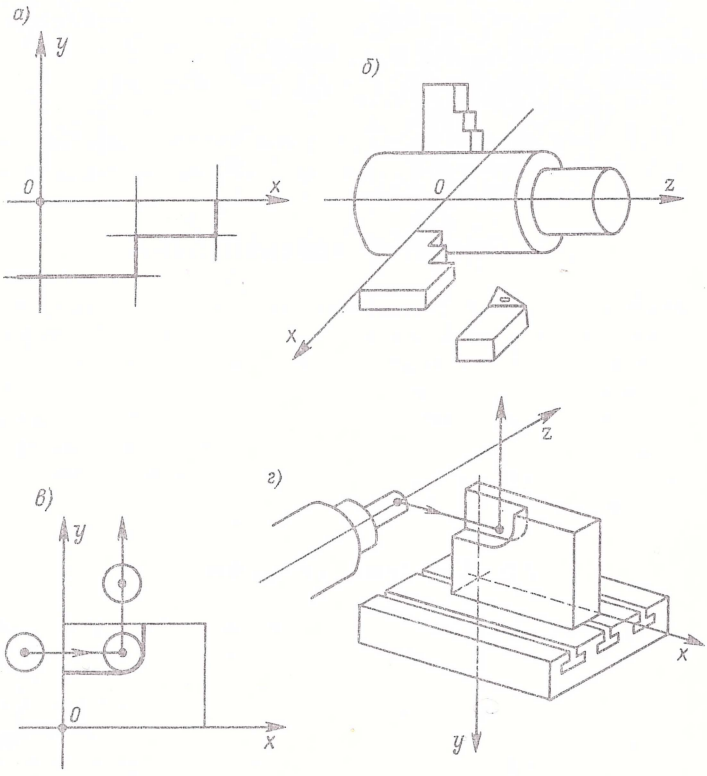


Рис. 2.3

2.9. Геометрические определения

Геометрическое определение является оператором присваивания, в левой части которого находится символическое имя определяемого геометрического объекта, а в правой — один из вариантов задания этого объекта в строгом соответствии с форматом, приведенным в данном разделе.

В общем виде геометрическое определение имеет следующий формат:

тип имя = список параметров

где *тип* — служебное слово, определяющее тип геометрического объекта; *имя* — символическое имя определяемого геометрического объекта.

Параметрами, определяющими геометрический объект, являются имена опорных элементов или вложенные геометрические определения, модификаторы, числа или

заменяющие их арифметические выражения (опорным элементом называют ранее определенный геометрический объект).

Служебными словами, определяющими тип геометрического объекта, являются ТОЧКА, ПРЯМАЯ, ОКРУЖН, ВЕКТОР, МАТР, ПЛОСК. Служебное слово, определяющее тип геометрического объекта, может отсутствовать, если тип объекта был указан ранее явно (отдельной строкой) или если используется символическое имя, определяющее тип геометрического объекта неявно.

2.9.1. Внутреннее представление геометрических объектов

Для внутреннего представления геометрических объектов в процессоре системы Техтран используются канонические формы (табл. 2.3). Внутреннее представление

Таблица 2.3

Тип объекта	Внутреннее представление
Точка	x, y, z — координаты точки
Прямая	$\cos\alpha$, $\sin\alpha$, p — коэффициенты нормального управления прямой ($x\cos\alpha + y\sin\alpha - p = 0$)
Окружность	x, y, r — координаты центра окружности и ее радиус
Вектор	Δx, Δy, Δz — проекции вектора на координатные оси
Плоскость	$\cos\alpha$, $\cos\beta$, $\cos\gamma$, p — коэффициенты нормального уравнения плоскости ($x\cos\alpha + y\cos\beta + z\cos\gamma - p = 0$)
Матрица	a_{11}, a_{12}, ..., a_{34} — коэффициенты нормальных уравнений координатных плоскостей системы координат, полученной при преобразовании исходной системы координат в соответствии с заданной матрицей

геометрических объектов применяют при выводе на печать результатов работы процессора в случае использования отладочных операторов СЛЕД и ПЧТ1 (см. п. 2.12.2). Для доступа к элементам канонической формы точек, прямых, окружностей и векторов следует использовать функцию КООРД. Например, требуется определить прямую, проходящую через центры двух окружностей, координаты которых неизвестны (рис. 2.4). Найдем эти координаты следующим образом:

```
X1 = КООРД (1, КР1)
У1 = КООРД (2, КР1)
X2 = КООРД (1, КР2)
У2 = КООРД (2, КР2)
```

Искомая прямая имеет вид

```
ПРА = X1, У1, X2, У2.
```

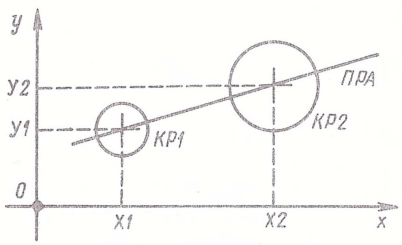


Рис. 2.4

2.9.2. Определение точки

В определениях точки используют следующие группы модификаторов:

1. Модификаторы выбора точки из двух возможных: ХБ — выбирается точка с большей координатой x, УБ — с большей координатой y, ХМ — с меньшей координатой x и УМ — с меньшей координатой y.

2. Модификаторы направления отсчета угла: ПОЧС — задается отсчет угла по часовой стрелке; ПРЧС — задается отсчет угла против часовой стрелки.

3. Модификаторы, указывающие координату точки, заданную в геометрическом определении: ХКООРД — координата x; УКООРД — координата y.

Точка, определенная прямоугольными координатами:

имя = координата x, координата y, координата z

имя = координата x, координата y

Например (рис. 2.5):
ТЧА = −5, 15, 10.7
ТОЧКА Т3 = 24, −8,2

Точка, определенная пересечением двух прямых:
имя = [ПЕРЕСЕЧ,] *прямая*$_1$, *прямая*$_2$
Например (рис. 2.6):
ТЧ2 = ПР1, ПР2

Точка, определенная пересечением прямой и окружности:
имя = [ПЕРЕСЕЧ,] *прямая*, *окружность*, {ХБ|УБ|ХМ|УМ}
Например (рис. 2.7):
ТОЧКА Т23 = [ПЕРЕСЕЧ,] ПР7, КР2, УБ

Примечание. Случай касания прямой и окружности рассматривается как пересечение этих двух объектов.

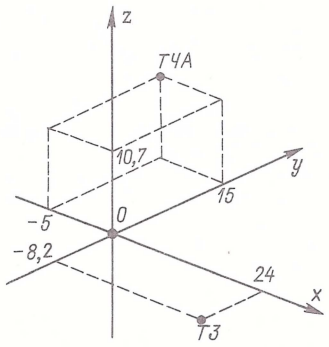


Рис. 2.5

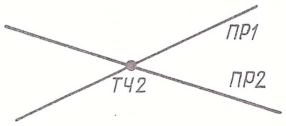


Рис. 2.6

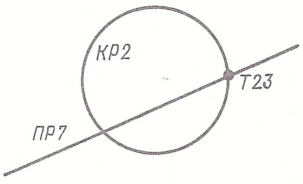


Рис. 2.7

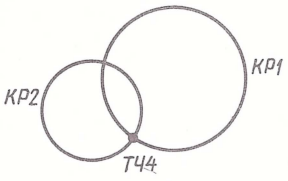


Рис. 2.8

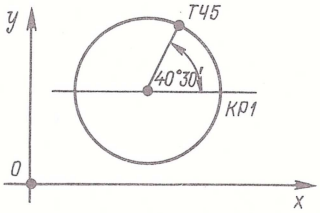


Рис. 2.9

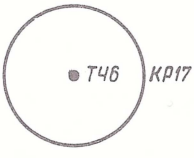


Рис. 2.10

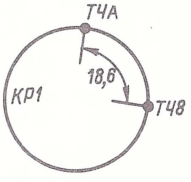


Рис. 2.11

Точка, определенная пересечением двух окружностей:
имя = [ПЕРЕСЕЧ,] *окружность*$_1$, *окружность*$_2$ {ХБ|УБ|ХМ|УМ}
Например (рис. 2.8):
ТЧ4 = КР1, КР2, УМ

Точка, определенная пересечением окружности и ее радиуса, проведенного под заданным углом к оси *x*:

имя = *окружность*, $\left\{\begin{matrix}\text{ПРЧС}\\\text{ПОЧС}\end{matrix}\right\}$, *угол*

Например (рис. 2.9):
ТЧ5 = КР1, ПРЧС, 40.5

Точка, являющаяся центром ранее определенной окружности:
имя = [ЦЕНТР,] *окружность*
Например (рис. 2.10):
ТЧ6 = КР17

Точка, находящаяся на окружности на известном расстоянии от другой точки этой же окружности:

имя = *окружность*, *точка* $\left\{\begin{matrix}\text{ПОЧС}\\\text{ПРЧС}\end{matrix}\right\}$, *расстояние*

Например (рис. 2.11):

ТЧ8=КР1, ТЧА, ПОЧС, 18.6

Примечание. Расстояние линейное, измеряется по дуге окружности.

Точка, находящаяся на прямой и имеющая заданную координату *x* или *y*:

имя=прямая, {ХКООРД|УКООРД}, *число*

Например (рис. 2.12):

ТЧ9=ПР1, ХКООРД, 80

Точка, находящаяся на прямой на определенном расстоянии от другой точки этой же прямой:

имя=прямая, *точка*, *расстояние*, {ХБ|УБ|ХМ|УМ}

Например (рис. 2.13):

ТЧ10=ПР8, ТЧА, 23, ХМ

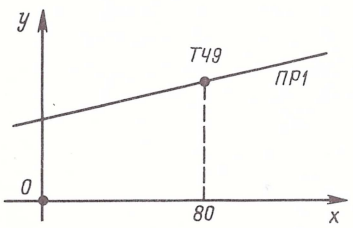


Рис. 2.12

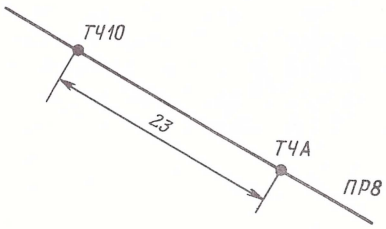


Рис. 2.13

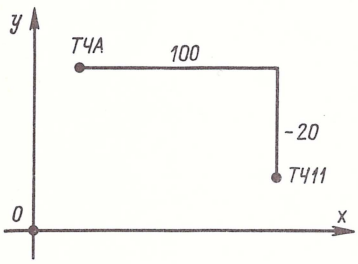


Рис. 2.14

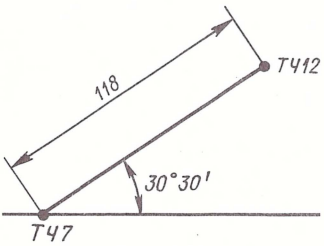


Рис. 2.15

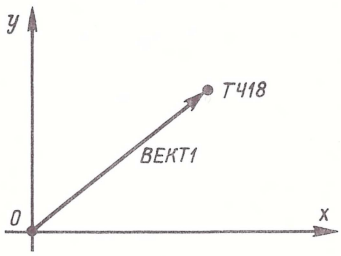


Рис. 2.16

Точка, заданная приращениями координат известной точки:

имя=точка, ПЕРЕНОС, *приращение x*, *приращение y*

Например (рис. 2.14):

ТЧ11=ТЧА, ПЕРЕНОС, 100, —20

Точка, находящаяся на определенном расстоянии от известной точки под углом к оси *x*:

имя=точка, $\left\{\begin{matrix}\text{ПОВХУ}\\\text{ПОВОРОТ}\end{matrix}\right\}$, *угол*, *расстояние*

Например (рис. 2.15):

ТЧ12=ТЧ7, ПОВХУ, 30.5, 118

Точка, координаты которой получены преобразованием координат известной точки в соответствии с указанной матрицей:

имя=точка, *матрица*

Например:

ТЧ16=ТЧА, МК6

Точка, определенная ранее заданной точкой:

имя=точка

Например:

ТЧ17=ТЧ13А

Точка, для которой ранее определенный вектор является радиус-вектором:

имя=вектор

Например (рис. 2.16):

ТЧ18=ВЕКТ1

2.9.3. Задание координаты z в геометрических определениях точек (оператор ПЛЗ)

Оператор ПЛЗ предназначен для задания координаты z тех точек, в геометрических определениях которых нет явно заданной третьей координаты:

Формат:

ПЛЗ $\sqcup$ { *число* / *имя* / ОТМЕН }

где *имя* — имя плоскости, определенной ранее; ОТМЕН — служебное слово, обозначающее отмену действия оператора ПЛЗ.

Действие оператора состоит в следующем. Если в операторах определения точки координата z не задана явно, то процессор вычислит ее исходя из того условия, что определяемая точка должна лежать в плоскости, заданной оператором ПЛЗ. Если оператор ПЛЗ не был задан или был выполнен оператор ПЛЗ ОТМЕН, то все координаты z точек, не заданные явно, принимают нулевое значение. Действие оператора ПЛЗ распространяется на все точки, определенные после него, до использования другого оператора ПЛЗ или оператора ПЛЗ ОТМЕН.

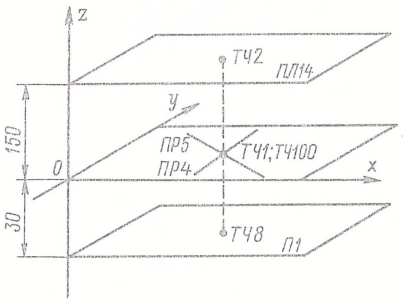


Рис. 2.17

На рис. 2.17 изображены точки, определенные в приведенном ниже фрагменте программы:

ТЧ1 = ПР5, ПР4
ПЛОСК П1, ПЛ14
ПЛ14 = 150
П1 = —30
ПЛЗ ПЛ14
ТЧ2 = ПР5, ПР4
ПЛЗ П1
ТЧ8 = ПР5, ПР4
ПЛЗ ОТМЕН
ТЧ100 = ПР5, ПР4

Координаты z точек ТЧ2 и ТЧ8 равны соответственно 150 и —30, так как они заданы оператором ПЛЗ с указанием плоскости, в которой лежит каждая из этих точек. Координата z точки ТЧ1 по умолчанию равна нулю; координата точки ТЧ100 также равна нулю, так как перед ее геометрическим определением задан оператор ПЛЗ ОТМЕН, отменяющий действие последнего оператора ПЛЗ.

2.9.4. Определение прямой

В определениях прямой используют следующие группы модификаторов:

1. Модификаторы, выбирающие одну прямую из двух, касающихся окружности: ХМ — выбирается прямая, точка касания которой имеет меньшую координату по x; ХБ — бóльшую координату по x; УМ — меньшую координату по y и УБ — бóльшую координату по y.

2. Модификаторы, указывающие, с какой стороны определяемая прямая касается окружности: СПРАВА — точка касания прямой находится справа от центра окружности, если смотреть в направлении от первого геометрического объекта правой части определения ко второму; СЛЕВА — прямая касается окружности слева (при тех же условиях рассмотрения геометрических объектов).

3. Модификаторы, указывающие положение прямой относительно другой: ПАРЛЕЛ — параллельно; ПЕРП — перпендикулярно.

Прямая, проходящая через две точки, заданные своими координатами:

имя = *координата* x_1, *координата* y_1, *координата* x_2, *координата* y_2

Например (рис. 2.18):

ПР20 = —15, 10, 35.6, 70

Прямая, проходящая через две точки:

имя = *точка*$_1$, *точка*$_2$

Например (рис. 2.19):

ПР21 = ТЧА, ТЧБ

Прямая, проходящая через точку и касающаяся окружности:

имя = *точка*, $\left\{\begin{matrix}\text{СЛЕВА}\\\text{СПРАВА}\end{matrix}\right\}$ [, КАСАТ], *окружность*

Например (рис. 2.20):

ПР22 = ТЧА, СПРАВА, КР3

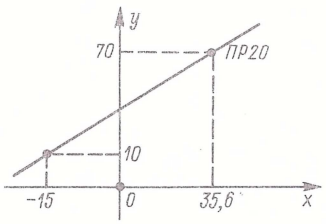


Рис. 2.18

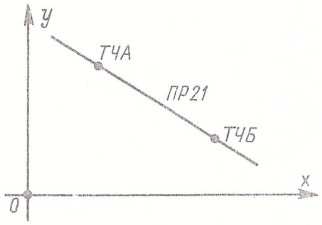


Рис. 2.19

Прямая, касающаяся двух окружностей:

имя = $\left\{\begin{matrix}\text{СПРАВА}\\\text{СЛЕВА}\end{matrix}\right\}$ [, КАСАТ], *окружность*$_1$, $\left\{\begin{matrix}\text{СПРАВА}\\\text{СЛЕВА}\end{matrix}\right\}$ [, КАСАТ], *окружность*$_2$

Например (рис. 2.21):

ПР23 = СЛЕВА, КР1, СПРАВА, КР3

Прямая, проходящая через точку под определенным углом к оси *x*:

имя = *точка*, *угол*

Например (рис. 2.22):

ПР23 = ТЧ5,— 15

Прямая, проходящая через точку под определенным углом к заданной прямой:

имя = *прямая*, *точка*, $\left\{\begin{matrix}\text{ПОВОРОТ}\\\text{ПОВХУ}\end{matrix}\right\}$, *угол*

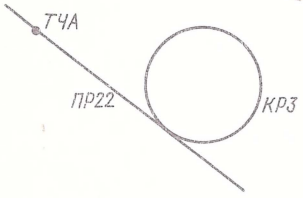


Рис. 2.20

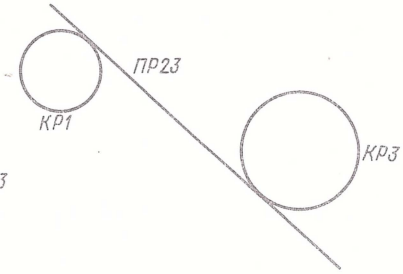


Рис. 2.21

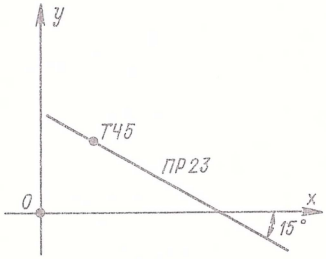


Рис. 2.22

Например (рис. 2.23):

ПР25 = ПРА, ТЧВ, 44.5

Прямая, проходящая через точку и перпендикулярная к другой прямой:

имя = *точка*, ПЕРП, *прямая*

Например (рис. 2.24):

ПР26 = ТЧ6, ПЕРП, ПР1

Прямая, проходящая через точку и параллельная другой прямой:

имя = *точка*, ПАРЛЕЛ, *прямая*

Например (рис. 2.25):

ПР27 = ТЧ1, ПАРЛЕЛ, ПР14

Прямая, проходящая параллельно другой прямой на заданном расстоянии от нее:
имя = [ПАРЛЕЛ,] *прямая, расстояние*, {ХБ|УБ|ХМ|УМ}
Например (рис. 2.26):
ПР28 = ПРА, 43,6, УМ

Прямая, касающаяся окружности и составляющая с осью *x* заданный угол:
имя = [КАС,] *окружность*, {ХБ|УБ|ХМ|УМ}, *угол*
Например (рис. 2.27):
ПР290 = КАС, КР1, УМ,— 15

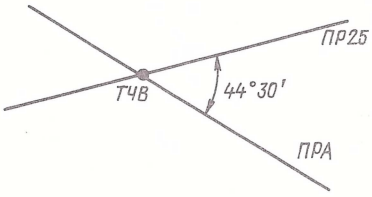


Рис. 2.23

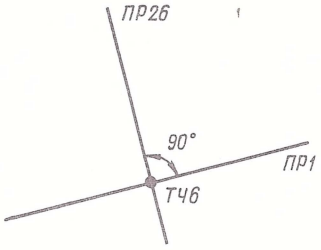


Рис. 2.24

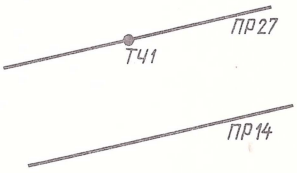


Рис. 2.25

Прямая, проходящая через точку параллельно вектору:
имя = *точка, вектор*
Например (рис. 2.28):
ПР310 = ТЧ1, ВЕК2

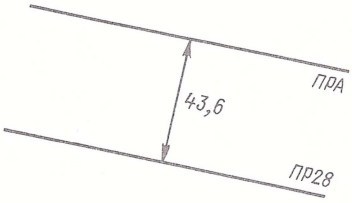


Рис. 2.26

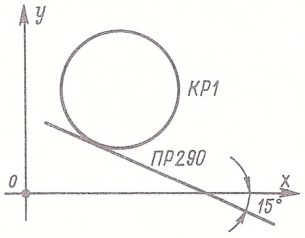


Рис. 2.27

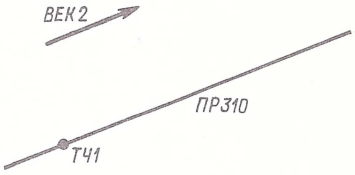


Рис. 2.28

Прямая, полученная преобразованием заданной прямой в соответствии с матрицей:
имя = *прямая, матрица*

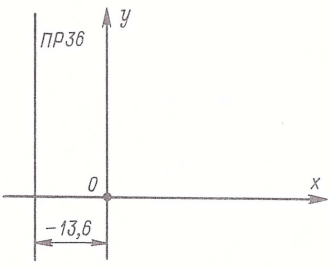


Рис. 2.29

Например:
ПР32 = ПР10, МАТР1

Прямая, совпадающая с прямой, определенной ранее:
имя = *прямая*
Например:
ПР13 = ПР376

Прямая, параллельная одной из осей координат и находящаяся на заданном расстоянии от нее:
имя = $\left\{\begin{matrix}\text{ХПАР}\\ \text{УПАР}\end{matrix}\right\}$, *расстояние*
Например (рис. 2.29):
ПР36 = УПАР, —13,6

2.9.5. Определение окружности

В определениях окружности используют следующие группы модификаторов:

1. Модификаторы БОЛШ и МЕНШ, служащие для выбора из двух окружностей одной — с бóльшим или меньшим радиусом соответственно.

2. Модификаторы, указывающие положение определяемой окружности относительно другой, являющейся опорным элементом: ВНЕ — окружность находится вне заданной окружности; ВНУТРИ — окружность находится внутри заданной окружности.

3. Модификаторы, указывающие положение окружности относительно прямой: ХМ — центр окружности находится слева от прямой, ХБ — справа от прямой, УБ — выше прямой и УМ — ниже прямой.

4. Те же модификаторы ХМ, ХБ, УБ, УМ, но служащие для выбора из двух окружностей одной по соответствующей координате ее центра.

Модификаторы 2 и 3-й групп относятся к следующим непосредственно за ними опорным элементам геометрического определения. Применение модификаторов ВНУТРИ и ВНЕ имеет свои особенности, отмеченные ниже.

Окружность, совпадающая с окружностью, определенной ранее:

имя = окружность

Например:

КР4 = КР100

Окружность, определенная координатами центра и радиусом:

имя = координата x, координата y, радиус

Например (рис. 2.30):

КР4 = — 13.6, 20, 5

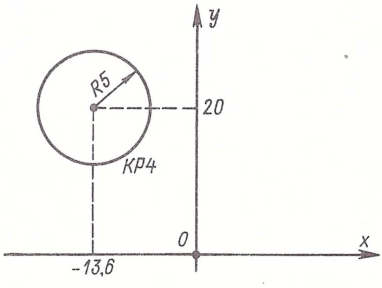


Рис. 2.30

Окружность, определенная радиусом и точкой, являющейся ее центром:

имя = [ЦЕНТР ,] *точка*, [РАДИУС,] *радиус*

Например (рис. 2.31):

КР4 = ТЧ3, 15

Окружность, определенная точкой, являющейся ее центром, и касающаяся заданной прямой:

имя = [ЦЕНТР,] *точка*, [КАС,] *прямая*

Например (рис. 2.32):

КР34 = ЦЕНТР, ТЧ6, КАС, ПР8

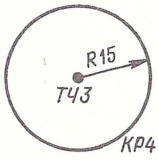


Рис. 2.31

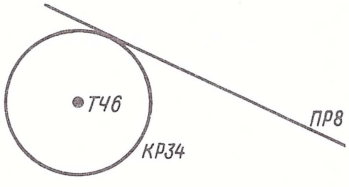


Рис. 2.32

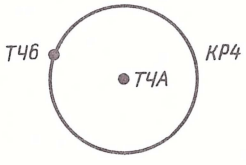


Рис. 2.33

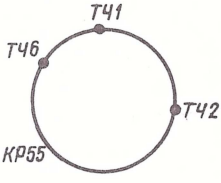


Рис. 2.34

Окружность, определенная точкой, являющейся ее центром, и проходящая через другую заданную точку:

имя = [ЦЕНТР,] $точка_1$, $точка_2$

Например (рис. 2.33):

КР4 = ТЧА, ТЧ6

Окружность, проходящая через три точки:

имя = $точка_1$, $точка_2$, $точка_3$

Например (рис. 2.34):

КР55 = ТЧ1, ТЧ6, ТЧ2

Окружность, определенная точкой, являющейся ее центром, и касающаяся другой окружности:

имя = [ЦЕНТР,] *точка*, {МЕНШ / БОЛШ}, [КАС ,] *окружность*

Например (рис. 2.35):
КР6 = ТЧ1, МЕНШ, КРА1
КР6А = ТЧ1, БОЛШ, КРА1

Окружность заданного радиуса, касающаяся двух пересекающихся прямых:

имя = {ХБ / УБ / ХМ / УМ}, *прямая*$_1$, {ХБ / УБ / ХМ / УМ}, *прямая*$_2$ [, РАДИУС], *радиус*

Например (рис. 2.36):
КР47 = УБ, ПР1, ХБ, ПР2, 18
КР7 = УМ, ПР1, ХБ, ПР2, 18

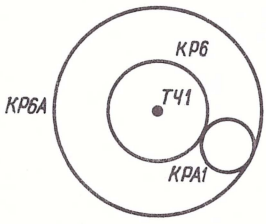


Рис. 2.35

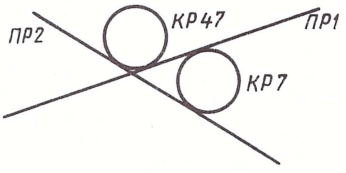


Рис. 2.36

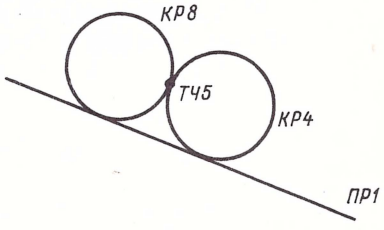


Рис. 2.37

Окружность заданного радиуса, касающаяся прямой и проходящая через заданную точку:

имя = [КАС,] *прямая*, {ХБ / ХМ / УБ / УМ}, *точка* [,РАДИУС], *радиус*

Например (рис. 2.37):
КР8 = ПР1, УБ, ТЧ5, 20
КР4 = ПР1, УМ, ТЧ5, 20

Окружность заданного радиуса, касающаяся прямой и окружности:

имя = {ХБ / УБ / ХМ / УМ}, *прямая*, {ХБ / ХМ / УБ / УМ}, {ВНЕ / ВНУТРИ}, *окружность*, *радиус*

Например (рис. 2.38):
КР1 = УБ, ПР1, ХМ, ВНЕ, КРО, 10
КР2 = УБ, ПР1, ХМ, ВНУТРИ, КРО, 10
КР3 = УБ, ПР1, ХБ, ВНУТРИ, КРО, 10
КР4 = УБ, ПР1, ХБ, ВНЕ, КРО, 10
КР5 = УМ, ПР1, ХМ, ВНЕ, КРО, 10
КР6 = УМ, ПР1, ХМ, ВНУТРИ, КРО, 10
КР7 = УМ, ПР1, ХБ, ВНУТРИ, КРО, 10
КР8 = УМ, ПР1, ХБ, ВНЕ, КРО, 10

Окружность заданного радиуса, касающаяся двух других окружностей:

имя = {ХБ / УБ / ХМ / УМ}, {ВНУТРИ / ВНЕ}, *окружность*$_1$, {ВНУТРИ / ВНЕ}, *окружность*$_2$, *радиус*

Например (рис. 2.39):
КР1 = УБ, ВНЕ, КРА, ВНЕ, КРБ, 15
КР2 = УБ, ВНЕ, КРА, ВНУТРИ, КРБ, 15
КР3 = УМ, ВНЕ, КРА, ВНУТРИ, КРБ, 15
КР4 = УМ, ВНЕ, КРА, ВНЕ, КРБ, 15
КР5 = УМ, ВНУТРИ, КРА, ВНЕ, КРБ, 15
КР6 = УБ, ВНУТРИ, КРА, ВНЕ, КРБ, 15
КР7 = УБ, ВНУТРИ, КРА, ВНУТРИ, КРБ, 15
КР8 = УМ, ВНУТРИ, КРА, ВНУТРИ, КРБ, 15

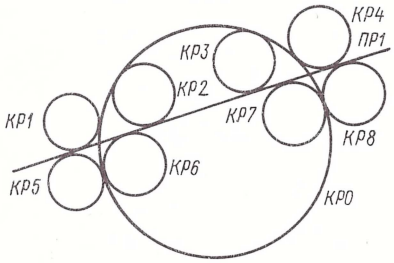


Рис. 2.38

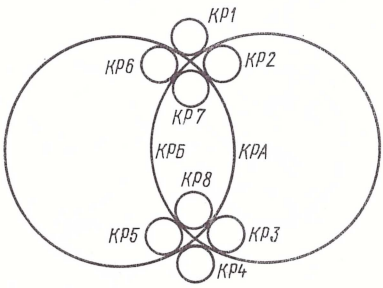


Рис. 2.39

Примечание. В тех случаях, когда определяемая окружность не может быть построена внутри опорных (слишком велик ее диаметр или опорные окружности не пересекаются), модификатор ВНУТРИ указывает положение опорных окружностей относительно определяемой окружности.

Например (рис. 2.40, *а*, *б*):
КР9 = УМ, ВНУТРИ, КР31, ВНУТРИ, КР40, 150
КР10 = УБ, ВНУТРИ, КР50, ВНЕ, КР17, 60
КР11 = УМ, ВНУТРИ, КР50, ВНУТРИ, КР17, 60

Окружность, касающаяся трех прямых:

имя = { ХБ / УБ / ХМ / УМ }, *прямая*$_1$, { ХБ / УБ / ХМ / УМ }, *прямая*$_2$, { ХБ / УБ / ХМ / УМ }, *прямая*$_3$

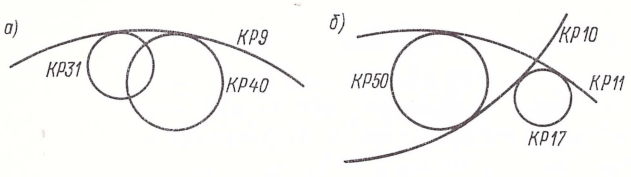


Рис. 2.40

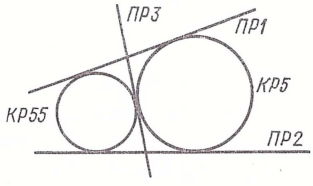


Рис. 2.41

Например (рис. 2.41):
КР5 = УМ, ПР1, УБ, ПР2, ХБ, ПР3
КР55 = УМ, ПР1, УБ, ПР2, ХМ, ПР3

Окружность заданного радиуса, проходящая через две точки:

имя = { ХБ / УБ / ХМ / УМ }, *точка*$_1$, *точка*$_2$ [,РАДИУС], *радиус*

Например (рис. 2.42):
КР1 = ХБ, ТЧ1, ТЧ2, 50
КР2 = ХМ, ТЧ1, ТЧ2, 30

Окружность заданного радиуса, проходящая через точку и касающаяся другой окружности:

имя = *точка*, {ХБ / УБ / ХМ / УМ}, {ВНУТРИ / ВНЕ}, *окружность* [, РАДИУС], *радиус*

Например (рис. 2.43):
КР3 = ТЧ1, УБ, ВНУТРИ, КРА, 25
КР5 = ТЧ1, УМ, ВНУТРИ, КРА, 25
КР6 = ТЧ9, УМ, ВНУТРИ, КРБ, 14
КР9 = ТЧ9, УМ, ВНЕ, КРБ, 14

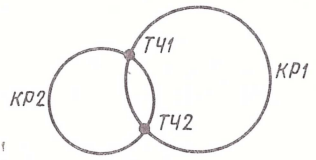


Рис. 2.42

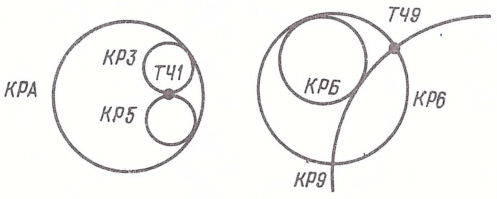


Рис. 2.43

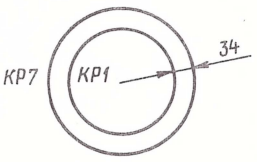


Рис. 2.44

Примечание. В том случае, когда точка, указанная в определении, находится внутри опорной окружности, использование модификатора ВНЕ не имеет смысла. Если опорная точка находится вне опорной окружности, возможно как внутреннее, так и внешнее касание. Модификатор ВНУТРИ в этом случае указывает положение опорной окружности относительно определяемой (окружность КР6 в приведенном примере).

Окружность, концентричная заданной окружности:

имя = ПАРЛЕЛ, {МЕНШ / БОЛШ}, *окружность*, *разность радиусов*

Например (рис. 2.44):
КР7 = ПАРЛЕЛ, КР1, БОЛШ, 34

Окружность, полученная преобразованием заданной окружности в соответствии с матрицей:

имя = *окружность*, *матрица*

Например:
КР7 = КР2, МАТР6

2.9.6. Определение вектора

В определениях вектора используют модификаторы, указывающие направление вектора: ХБ — вектор направлен в сторону возрастания координаты x; ХМ — в сторону уменьшения координаты x; УБ — в сторону возрастания координаты y и УМ — вектор направлен в сторону уменьшения координаты y.

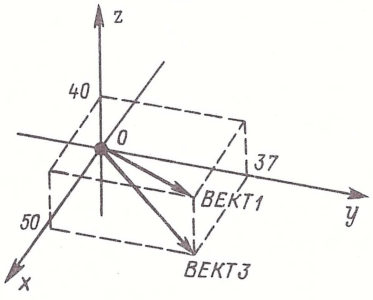


Рис. 2.45

Вектор, определенный проекциями на оси, начало которого находится в начале координат:

имя = *проекция x*, *проекция y* [,*проекция z*]

Например (рис. 2.45):
ВЕКТ1 = 50, 37, 40
ВЕКТ3 = 50, 37

Вектор, определенный своими начальной и конечной точками:

имя = *точка*$_1$, *точка*$_2$

Например (рис. 2.46):
ВКТР = ТЧ1, ТЧ2

Вектор, равный сумме двух других векторов:

имя = *вектор*$_1$, ПЛЮС, *вектор*$_2$

Например (рис. 2.47):
ВЕКТ4 = ВЕКР1, ПЛЮС, ВКТ2

Вектор, равный разности двух векторов:
имя = *вектор*$_1$, МИНУС, *вектор*$_2$
Например (рис. 2.48):
ВЕКТ5 = ВЕКР1, МИНУС, ВКТ2

Вектор, равный произведению заданного вектора и числа:
имя = *число*, *вектор*
Например (рис. 2.49):
ВЕКТ7 = 3, ВЕ2

Вектор, равный произведению двух векторов:
имя = *вектор*$_1$, УМНОЖ, *вектор*$_2$
Например:
ВЕКТ11 = ВА36, УМНОЖ, ВР7

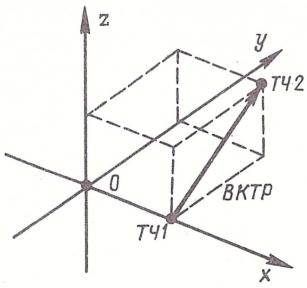


Рис. 2.46

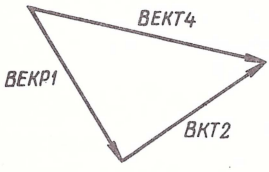


Рис. 2.47

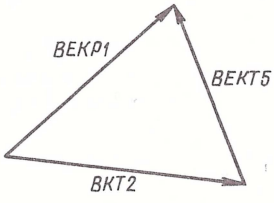


Рис. 2.48

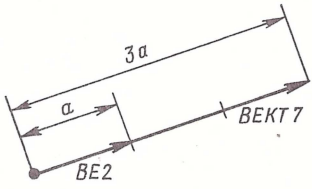


Рис. 2.49

Вектор, равный ранее определенному вектору:
имя = *вектор*
Например:
ВЕР4 = ВКР6

Вектор, полученный преобразованием ранее определенного вектора в соответствии с заданной матрицей:
имя = *вектор*, *матрица*
Например:
ВЕКТОР В6 = ВЕКТ, М2

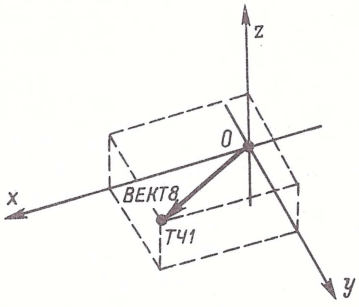


Рис. 2.50

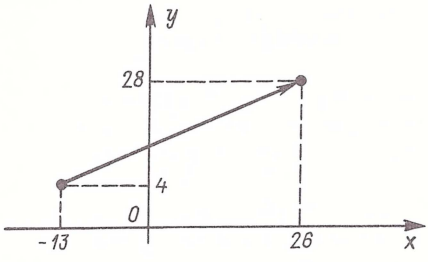


Рис. 2.51

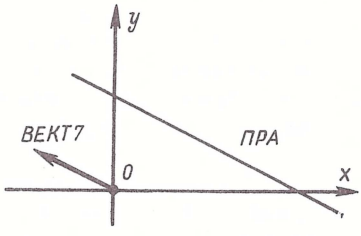


Рис. 2.52

Вектор, являющийся радиус-вектором заданной точки:
имя = *точка*
Например (рис. 2.50):
ВЕКТ8 = ТЧ1

Вектор, заданный координатами своих начальной и конечной точек:
имя = *координата* x_1, *координата* y_1, *координата* x_2, *координата* y_2:
Например (рис. 2.51):
ВЕКТ В1 = — 13, 4, 26, 28

Вектор, параллельный заданной прямой:

имя=*прямая*, {ХБ|УБ|ХМ|УМ}

Примечание. Длина вектора равна единице, начало его находится в начале координат, например (рис. 2.52):

ВЕКТ7=ПРА, ХМ

2.9.7. Определение матрицы

Матрица определяет способ преобразования координат. Это преобразование может быть параллельным переносом исходной системы координат, ее поворотом, симметрией или комбинацией этих преобразований. Любое преобразование координатной системы может быть описано с помощью системы трех уравнений:

$$a_{11}x_1 + a_{12}y_1 + a_{13}z_1 + a_{14} = x_2$$
$$a_{21}x_1 + a_{22}y_1 + a_{23}z_1 + a_{24} = y_2$$
$$a_{31}x_1 + a_{32}y_1 + a_{33}z_1 + a_{34} = z_2$$

Всегда можно определить значения таких коэффициентов a_{11}, a_{12}, ..., a_{34}, что при подстановке в систему уравнений значений координат произвольной точки x_1, y_1, z_1 в локальной системе координат будут получены координаты этой точки x_2, y_2, z_2 в абсолютной (исходной) системе координат.

Коэффициенты данной системы уравнений используются для представления матрицы преобразования в канонической форме и выводятся на печать в следующей последовательности:

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{14} & a_{24} & a_{34} \end{matrix}$$

Первые три элемента каждого столбца определяют направление осей локальной системы координат по отношению к исходной (абсолютной) системе координат: a_{11}, a_{21}, a_{31} — единичный вектор оси x_1; a_{12}, a_{22}, a_{32} — единичный вектор оси y_1; a_{13}, a_{23}, a_{33} — единичный вектор оси z_1.

Четвертая строка определяет положение начала координат локальной системы: a_{14}, a_{24}, a_{34} — координаты x, y, z начала локальной системы координат в исходной (абсолютной) системе.

Результатом матричного преобразования является геометрический объект, который имеет в локальной системе координат, полученной преобразованием исходной системы, параметры исходного геометрического объекта.

При использовании комбинированных матричных преобразований, задаваемых произведением матриц, следует помнить, что матричное умножение не коммутативно, т. е. конечный результат зависит от последовательности элементарных преобразований.

На рис. 2.53, *а—в* представлено последовательное преобразование исходной системы координат (*а*) матрицами поворота (*б*) и переноса (*в*), а на рис. 2.53, *г—е* дано преобразование системы матрицами переноса (*д*) и поворота (*е*).

В геометрических определениях матрицы используются следующие модификаторы: ПОВХУ — определяет поворот осей x и y вокруг оси z; ПОВУЗ — поворот осей y и z вокруг оси x и ПОВЗХ — поворот осей z и x вокруг оси y (направление отсчета углов поворота — в соответствии с рисунком).

Матрица, определяющая координатную систему, полученную поворотом на заданный угол исходной системы координат вокруг одной из ее осей:

имя={ПОВХУ|ПОВУЗ|ПОВЗХ}, *угол*

Например (рис. 2.54):

МАТР МТ2=ПОВЗХ, 30.5

Матрица, определяющая координатную систему, полученную параллельным переносом исходной системы координат:

имя = ПЕРЕНОС, *координата x*, *координата y* [, *координата z*]

Например (рис. 2.55):

МАТР3 = ПЕРЕНОС, 10, 15, 20

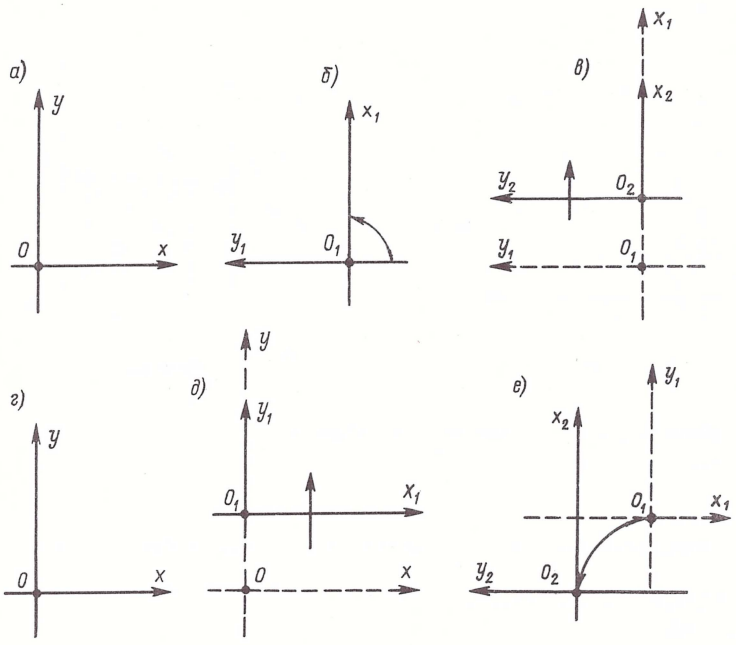


Рис. 2.53

Матрица, определяющая координатную систему, полученную параллельным переносом исходной системы координат и последующим поворотом на заданный угол вокруг одной из координатных осей:

имя = { ПОВХУ / ПОВУЗ / ПОВЗХ }, *угол*, ПЕРЕНОС, *координата x*, *координата y*, *координата z*

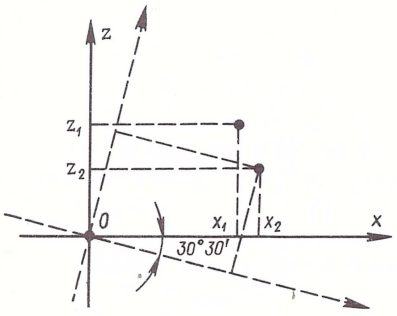


Рис. 2.54

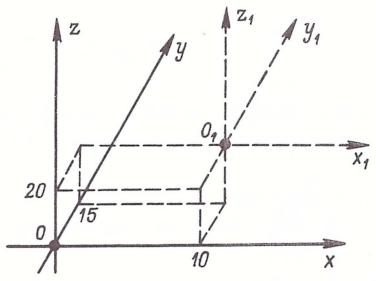


Рис. 2.55

Например (рис. 2.56):

МАТР = ПОВЗХ, 30.5, ПЕРЕНОС, 10, 15, 20

Матрица, являющаяся произведением двух других матриц:

имя = *матрица*$_1$, *матрица*$_2$

Например (рис. 2.56):
МАТР = МАТР3, МТ2

Примечания: 1. Произведение матриц эквивалентно применению сначала *матрицы*$_2$, а затем преобразованию полученного результата в соответствии с *матрицей*$_1$. 2. Матрица, заданная в примере (см. рис. 2.56), является произведением матриц, определенных в первых двух примерах (см. рис. 2.54 и 2.55).

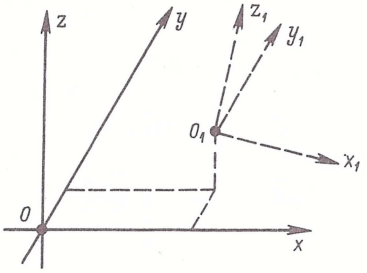


Рис. 2.56

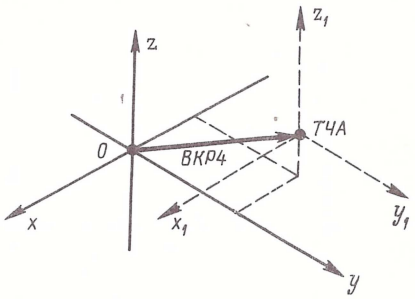


Рис. 2.57

Матрица, равная ранее определенной матрице:
имя = *матрица*
Например:
МАТР4 = МТ6

Матрица, определяющая перенос начала координат в точку, заданную явно или вектором:
имя = ПЕРЕНОС, {*точка*|*вектор*}
Например (рис. 2.57):
МКТ = ПЕРЕНОС, ТЧА
МКТ = ПЕРЕНОС, ВКР4

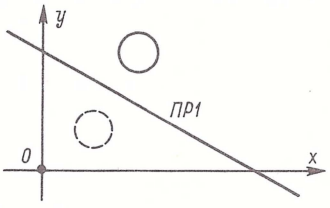


Рис. 2.58

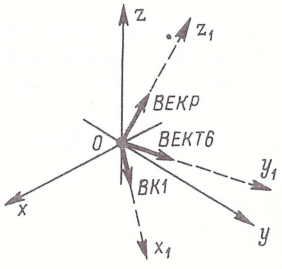


Рис. 2.59

Матрица, определяющая поворот исходной системы координат вокруг одной из осей и последующий перенос начала координат в точку, заданную явно или вектором:
имя = {*точка*|*вектор*}, {ПОВХУ|ПОВУЗ|ПОВЗХ}, *угол*

Матрица, определяющая симметрию относительно заданной прямой:
имя = *прямая*
Например (рис. 2.58):
МАТР8 = ПР1

Матрица, определяющая координатную систему, начало которой совпадает с началом координат исходной системы, а направления осей заданы векторами:
имя = *вектор*$_1$, *вектор*$_2$, *вектор*$_3$
Например (рис. 2.59):
МАТР9 = ВК1, ВЕКТ6, ВЕКР

Матрица, определяющая координатную систему, начало которой находится в точке, заданной явно или вектором, а направление осей заданы соответствующими векторами:

имя = *вектор*$_1$, *вектор*$_2$, *вектор*$_3$, ПЕРЕНОС, {*точка*|*вектор*$_4$}

Например (рис. 2.60):

М2 = ВК4, ВКТ, В16, ПЕРЕНОС, ТЧА

Матрица, определяющая координатную систему, являющуюся зеркальным отображением исходной относительно заданной плоскости:

имя = *плоскость*

Например (рис. 2.61):

МАТР2 = ПЛОСК4

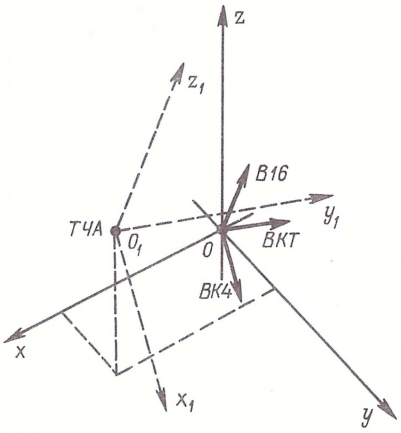


Рис. 2.60

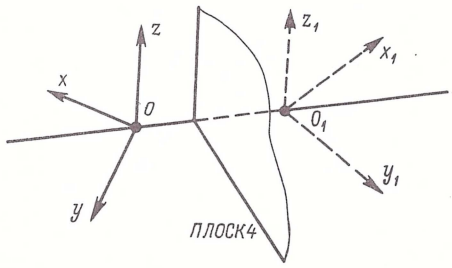


Рис. 2.61

Матрица, определяющая преобразование координат, обратное преобразованию, определенному заданной матрицей:

имя = ОБРАТН, *матрица*

Например; если ТЧ2 = МАТР2, ТЧ1 и М12 = ОБРАТН, МАТР2, то справедливо геометрическое определение: ТЧ1 = ТЧ2, М12

2.9.8. Определение плоскости

В геометрических определениях плоскости используются следующие модификаторы: ХУПАР — определяемая плоскость параллельна плоскости *xy*, УЗПАР — параллельна плоскости *yz* и ЗХПАР — плоскости *zx*. Отсутствие модификатора эквивалентно использованию модификатора ХУПАР.

Плоскость, параллельная одной из координатных плоскостей и проходящая через заданную точку:

имя = {ХУПАР|УЗПАР|ЗХПАР}, *точка*

Например:

ПЛОСК П1 = УЗПАР, ТЧ3

Плоскость, параллельная одной из координатных плоскостей и находящаяся на заданном расстоянии от нее:

имя = {ХУПАР|УЗПАР|ХЗПАР}, *число*

Например:

ПЛОСК8 = ЗХПАР, —29.6

Плоскость, совпадающая с плоскостью, определенной ранее:

имя = *плоскость*

Например:

ПЛОСК ПЛТ = ПЛОСК8

Плоскость, определенная коэффициентами ее нормального уравнения:

имя = *число*$_1$, *число*$_2$, *число*$_3$, *число*$_4$

Плоскость, проходящая через три точки, не лежащие на одной прямой:

имя = *точка*$_1$, *точка*$_2$, *точка*$_3$

Например:

ПЛ54 = ТЧ1, ТЧ18, ТЧ46

Плоскость, проходящая через заданную точку перпендикулярно вектору:

имя = *вектор*, *точка*

Например (рис. 2.62):

ПЛОСК8 = ВЕКР, ТЧ2

Плоскость, параллельная двум векторам и проходящая через заданную точку:

имя = *вектор*$_1$, *вектор*$_2$, *точка*

Например (рис. 2.63):

ПЛ6 = ВЕКТ1, ВЕКИ2, ТЧ1

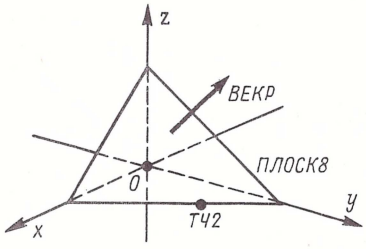


Рис. 2.62

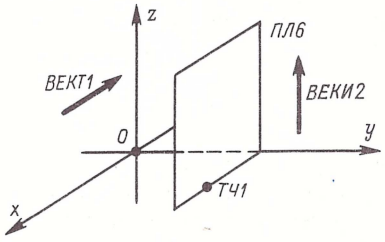


Рис. 2.63

Плоскость, параллельная другой плоскости и проходящая через заданную точку:

имя = *точка*, ПАРЛЕЛ, *плоскость*

Плоскость, полученная преобразованием заданной плоскости в соответствии с указанной матрицей:

имя = *плоскость*, *матрица*

2.9.9. Вложенное определение

Вместо имени ранее определенного геометрического объекта в правой части геометрического определения или в операторе движения можно задать «вложенное определение». Оно состоит из типа определяемого объекта, необязательного имени объекта и заключенного в скобки варианта геометрического определения соответствующего типа.

Если у такого геометрического объекта отсутствует имя, то значения его геометрических параметров сохраняются временно и используются процессором только для промежуточных вычислений.

На рис. 2.64 изображена окружность КР2, которая описана с использованием вложенных определений:

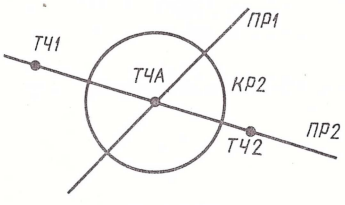


Рис. 2.64

КР2 = ТОЧКА (ПРЯМАЯ ПР2 (ТЧ1, ТЧ2), ПР1), РАДИУС, 2.5

Данное геометрическое определение равноценно трем другим определениям:

ПР2 = ТЧ1, ТЧ2

ТЧА = ПР2, ПР1

КР2 = ЦЕНТР, ТЧА, РАДИУС, 2.5

Различие состоит лишь в том, что значение точки ТЧА во втором варианте записи сохранится, а при использовании вложенного определения точка пересечения прямых ПР1 и ПР2 не может быть использована в программе повторно. Глубину вложения можно увеличивать. В каждом из геометрических определений она может достигнуть двадцати.

2.10. Программирование движений инструмента

2.10.1. Общие сведения

В языке Техтран существуют два типа движения:

1) движение *точка—точка* (поточечное движение), при котором перемещение инструмента программируется прямым заданием либо целевой точки, либо величины смещения;

2) непрерывное движение, при котором перемещение программируется перечислением элементов траектории инструмента (т. е. участков прямых, окружностей, частей контуров) с указанием направления движения.

Возможность управления поверхностью детали позволяет программировать обработку контуров, имеющих наклонное «дно», благодаря чему с помощью языка Техтран возможно (с определенными ограничениями) описывать обработку деталей достаточно сложной пространственной формы. Во всех операторах движения вместо имен геометрических объектов можно использовать вложенные определения объектов тех же типов.

2.10.2. Исходное положение инструмента

Прежде чем программировать движение инструмента, необходимо задать его исходное положение. Для этого служит оператор ИЗ.

Оператор имеет следующий формат:

ИЗ $\left\{\begin{matrix} точка \\ x, y\ \ [, z] \end{matrix}\right\}$

где *точка* — имя точки; x, y, z — координаты точки.

В качестве координат можно использовать арифметические выражения. Если отсутствует координата z, то она считается равной нулю. В операторе ИЗ допускается использовать вложенные определения. В одной программе может присутствовать несколько операторов ИЗ.

Например, возможны следующие варианты оператора ИЗ:

```
ИЗ 10, 20
ИЗ A, B
ИЗ (x+y)/2, (x—y)/2
ИЗ ТЧ1
ИЗ ТОЧКА (ПР1, ПР2)
```

2.10.3. Движение *точка — точка*

При использовании этого оператора программист явно указывает каждое перемещение инструмента либо приращениями относительно предыдущего положения, либо точкой результирующего положения центра (вершины) инструмента. Этот тип движения реализуется операторами ВТОЧКУ и ПРИРАЩ. В дальнейшем этот тип движения в некоторых случаях будет именоваться как «поточечное движение».

Оператор ВТОЧКУ вызывает перемещение в заданную точку и имеет формат:

ВТОЧКУ $\left\{\begin{matrix} точка \\ x, y\ \ [, z] \end{matrix}\right\}$

где *точка* — имя точки; x, y, z — координаты точки.

Вместо чисел в качестве координат точки можно использовать арифметические выражения. Если в операторе указаны только две координаты, то координата z считается равной нулю. На рис. 2.65 приведен пример движения *точка — точка*:

```
ТЧ3=35, 10
ИЗ 0, 0
ВТОЧКУ 0, —28
ВТОЧКУ 35, 10, 12
ВТОЧКУ ТЧ3
```

Оператором ПРИРАЩ указывается перемещение относительно текущего положения на заданный вектор. Оператор имеет следующий формат:

ПРИРАЩ $\left\{ \begin{array}{l} вектор \\ x, y\ [, z] \\ z \end{array} \right\}$

где *вектор* — имя вектора; x, y, z — проекции вектора на координатные оси.

Вместо чисел в качестве проекций можно использовать арифметические выражения. Если в операторе ПРИРАЩ координаты x и y или координата z равны нулю, то перемещений по соответствующим осям не происходит. Поэтому оператор ПРИРАЩ 0, 0, А эквивалентен оператору ПРИРАЩ А.

При многократном использовании оператора ПРИРАЩ следует соблюдать осторожность, так как ошибка в начале последовательности влияет на остальные положения инструмента.

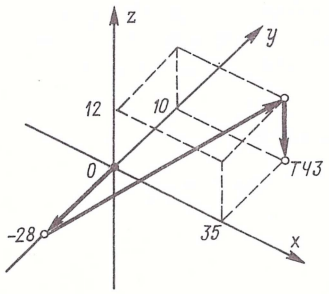


Рис. 2.65

Движение, изображенное на рис. 2.65, можно описать при помощи оператора ПРИРАЩ следующим образом:

```
ИЗ 0, 0
ПРИРАЩ 0, —28, 0
ПРИРАЩ 35, (28+10), 12
ПРИРАЩ —12
```

При описании движения допускается сочетание операторов ВТОЧКУ и ПРИРАЩ:

```
ИЗ 0, 0
ПРИРАЩ 0, —28, 0
ВТОЧКУ 35, 10, 12
ПРИРАЩ —12
```

Сокращенная запись операторов поточечного движения — это замена одним оператором нескольких записанных подряд операторов поточечного движения. Операнды этих операторов разделяются в этом случае двоеточием. Можно чередовать различные формы задания точек (приращения), например:

```
ВТОЧКУ ТЧ1: 6.2, А+В, 0 : ТОЧКА (ПР1, ПР2) : ТЧ3
ПРИРАЩ 5,1 : 0,3*КОС (45) : ВЕКТ1
```

Следует иметь в виду, что сокращенная запись этих операторов движения затрудняет понимание диагностических сообщений, имеющих ссылку на номер строки.

2.10.4. Непрерывное движение

При программировании непрерывного движения на языке Техтран описывается движение инструмента относительно трех поверхностей (рис. 2.66, *а*): поверхности детали (*ПД*); направляющей поверхности (*НП*); ограничивающей поверхности (*ОП*). На рис. 2.66, *б* приведено взаимное положение инструмента и поверхности. Рабочая часть инструмента при этом находится в постоянном контакте с поверхностью детали и направляющей поверхностью. Конечное положение инструмента после выполнения очередного оператора непрерывного движения определяется ограничивающей поверхностью (рис. 2.67, *а*). Прямые и окружности, составляющие контур детали, представляют собой проекции *НП* и *ОП* на горизонтальную плоскость (рис. 2.67, *б*).

Поверхность детали представляет собой плоскость — горизонтальную или наклонную. По умолчанию *ПД* совпадает с плоскостью *xOy*. В дальнейшем для простоты изложения *НП* и *ОП* отождествляются со своими проекциями на плоскость *xOy*.

Оператор ИНСТР (*описание инструмента*) предназначен для задания геометрических параметров инструмента, необходимых для расчета траектории его движения относительно контура и поверхности детали. Оператор имеет следующий формат:

ИНСТР *диаметр* $\left[, \left\{ \begin{array}{c} 0 \\ радиус \end{array} \right\} \right]$

где *диаметр* — диаметр инструмента; *радиус* — радиус скругления режущей кромки.

Радиус скругления не может быть больше радиуса цилиндрической части фрезы. Отсутствие оператора ИНСТР эквивалентно заданию инструмента нулевого диаметра. Новый оператор ИНСТР отменяет действие предыдущего. При непрерывном движении изменение диаметра инструмента запрещено.

На рис. 2.68 и 2.69 изображены различные резцы и фрезы и их описание на языке Техтран. На рис. 2.70, *а*, *б* представлены схемы электроэрозионной обработки и формирования зоны резания и описания на языке Техтран (A — ширина реза, $A=d+2\delta$). Точкой отмечен геометрический центр инструмента. Траекторией движения инструмента является линия, по которой перемещается в пространстве геометрический центр инструмента.

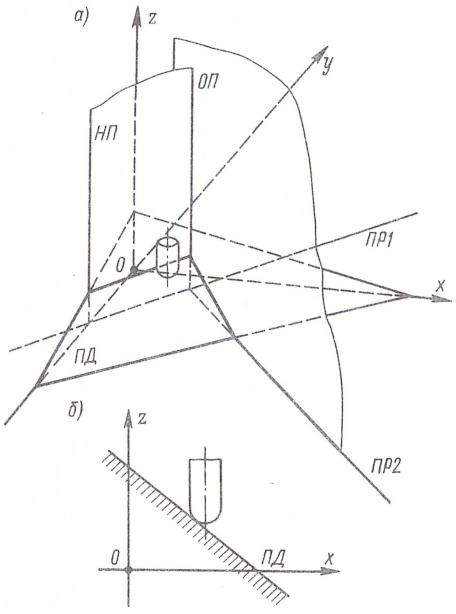


Рис. 2.66

Оператор МЕТОД используют для задания метода интерполяции и режима аппроксимации окружности отрезками прямой.

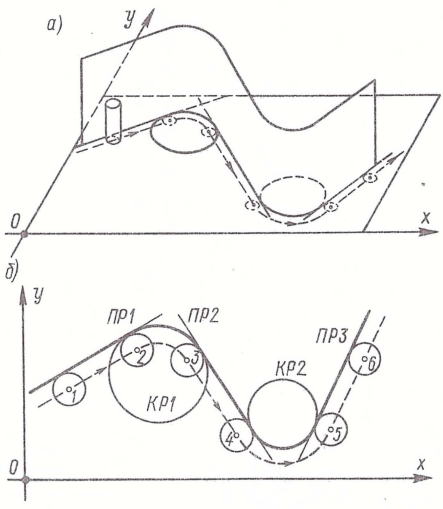


Рис. 2.67

Формат оператора:

$$\text{МЕТОД}\left\{\begin{array}{l}\text{ЛИНЕЙН}\quad[,\text{МАКСЛИН},\ \textit{число}_1]\\ \text{ЛИНКРУГ}\left\{\begin{array}{l}[,\text{КВАДР}]\\ ,\text{НЕКВАДР}\end{array}\right\}[,\text{МАКСРАД},\ \textit{число}_2]\end{array}\right\}$$

где ЛИНЕЙН — линейная интерполяция (аппроксимация окружностей отрезками прямых); ЛИНКРУГ — линейно-круговая интерполяция (аппроксимация кривых дугами окружностей); КВАДР — линейно-круговая интерполяция с делением окружностей на квадранты; НЕКВАДР — то же без деления на квадранты; $число_1$ — максимальная длина отрезков, аппроксимирующих окружность; $число_2$ — максимальный радиус окружности, при котором автоматически происходит переход к линейной интерполяции.

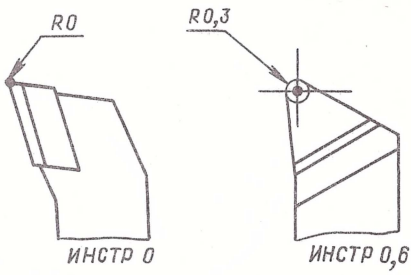


Рис. 2.68

На рис. 2.71, *а*, *б* представлены линейная (*а*) и линейно-круговая (*б*) интерполяции.

Параметр МАКСЛИН применяется для ограничения длины аппроксимирующих отрезков (рис. 2.72):

МЕТОД ЛИНЕЙН, МАКСЛИН, 4, 35.

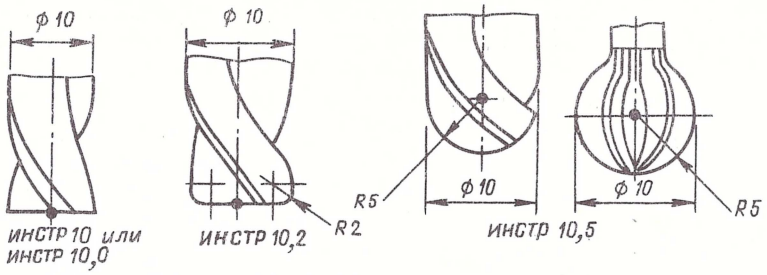


Рис. 2.69

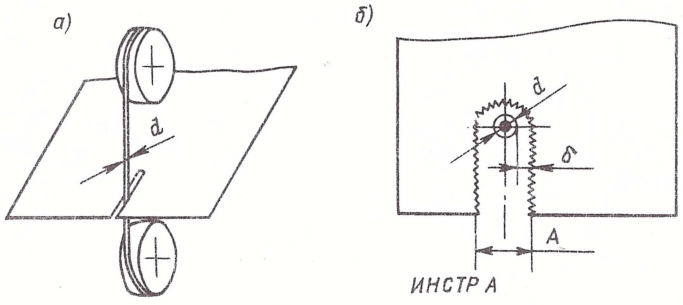


Рис. 2.70

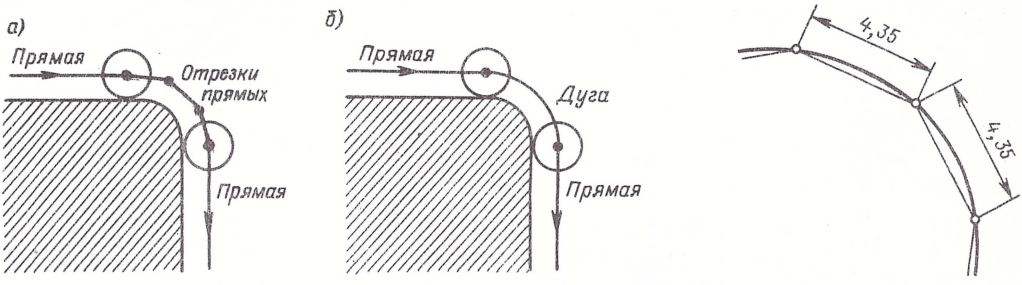


Рис. 2.71

Рис. 2.72

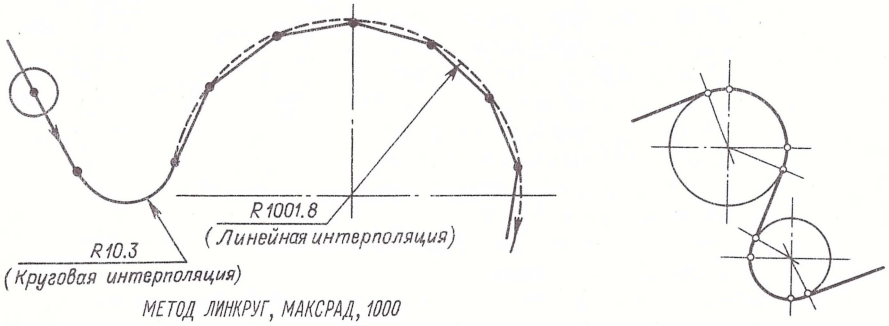


Рис. 2.73

Рис. 2.74

На рис. 2.73 показано применение параметра МАКСРАД в операторе МЕТОД. Точками ограничены участки траектории, заданные в управляющей программе отдельными кадрами:

МЕТОД ЛИНКРУГ, МАКСРАД, 1000.

При использовании модификатора КВАДР процессор системы разбивает дуги окружностей, составляющих траекторию, на участки, лежащие в отдельных квадрантах (рис. 2.74).

Отсутствие в программе оператора МЕТОД эквивалентно применению оператора МЕТОД ЛИНКРУГ, КВАДР. Круговая интерполяция разрешена только в одной из координатных плоскостей или в плоскостях, параллельных им. Круговая интерполяция с делением на квадранты возможна только в плоскостях, параллельных плоскости *xOy*.

Если поверхность детали не параллельна плоскости *xOy*, следует задавать оператор МЕТОД, ЛИНЕЙН, даже если контур не содержит дуг окружностей.

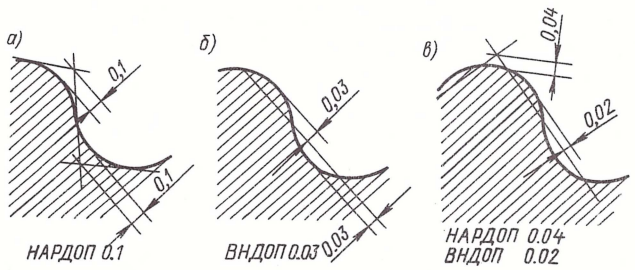


Рис. 2.75

Операторы НАРДОП *и* ВНДОП используются для задания точности линейной аппроксимации дуг окружности, которые заменяются отрезками прямых. НАРДОП *число* — точность аппроксимации касательными — наружный допуск (рис. 2.75, *а*); ВНДОП *число* — точность аппроксимации хордами — внутренний допуск (рис. 2.75, *б*).

При одновременном использовании этих операторов производится аппроксимация секущими (рис. 2.75, *в*). Если не заданы ни НАРДОП, ни ВНДОП, то система устанавливает следующие их значения:

НАРДОП.0, 05

ВНДОП 0

Оператор ИДИ задает начало непрерывного движения. Оператор ИДИ выводит инструмент в рабочее положение относительно управляющих поверхностей *НП*, *ПД* и *ОП*. Положение инструмента относительно каждой из них указывается модификаторами ДО, НА, ЗА. Оператор имеет следующий формат:

$$\text{ИДИ}\ \left\{\begin{matrix}\text{ДО}\\ \text{НА}\\ \text{ЗА}\end{matrix}\right\}\ НП\ \left[\left\{\begin{matrix}\text{ДО}\\ \text{НА}\end{matrix}\right\}\ ПД\right]\left[\left\{\begin{matrix}\text{ДО}\\ \text{НА}\\ \text{ЗА}\end{matrix}\right\}\ ОП\right]$$

По команде ИДИ инструмент выводится в требуемую точку по кратчайшему пути. Если *НП* и *ОП* имеют две точки пересечения, то выбирается ближайшая из них. На рис. 2.76 показано применение модификаторов ДО, НА, ЗА при движении с одной (*а*—*в*) и с двумя поверхностями (*г*—*д*) соответственно, а иллюстрация этих примеров на языке Техтран приведена ниже:

ИЗ ТЧ1; ИДИ ДО ПР1 (*а*)

ИЗ ТЧ1; ИДИ НА ПР1 (*б*)

ИЗ ТЧ1; ИДИ ЗА ПР1 (*в*)

ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПР2 (*г*)

ИЗ ТЧ1; ИДИ ДО ПР1 ЗА ПР2 (*д*)

При необходимости направление подхода может указываться операторами НАПРВ (в направлении вектора) или НАПРТ (в направлении точки).

На рис. 2.77, *а*, *б* приведены соответственно иллюстрации операторов:

ИДИ НА ПРХ ДО ПЛ1 ЗА ПР3

ИДИ НА ПРХ НА ПЛ1 ЗА ПР3

Если в операторе ИДИ задана поверхность детали, то при непрерывном движении инструмент будет находиться в контакте с ней в соответствии с заданным моди-

фикатором. Команды поточечного движения (ВТОЧКУ и ПРИРАЩ), вызывающие отход от поверхности детали по координате z, будут игнорироваться. Отвести инструмент от *ПД* можно только после оператора ПОВДЕТ ОТМЕН.

Если же *ПД* в операторе ИДИ опущена, то поверхностью детали становится плоскость, параллельная плоскости xOy и отстоящая от нее на значение текущей координаты z. При этом разрешено перемещать инструмент по вертикальной оси.

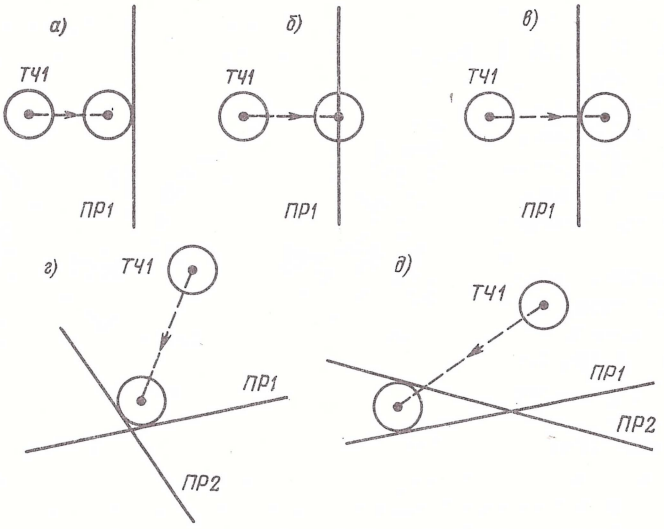


Рис. 2.76

Различие в использовании оператора ИДИ с одной, двумя и тремя поверхностями иллюстрирует рис. 2.78:

перемещение *0—1*:
ИЗ ТЧ1; ИДИ ДО ПР1
перемещение *0—2*:
ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПР2
перемещение *0—3*:
ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПЛ1
перемещение *0—4*:
ИЗ ТЧ1; ИДИ ДО ПР1 ДО ПЛ1 ДО ПР2

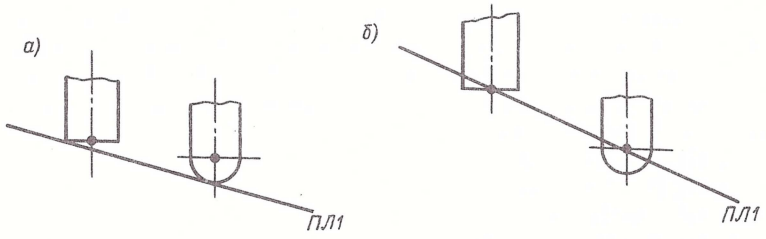


Рис. 2.77

В операторе ИДИ с двумя или тремя поверхностями в качестве *ОП* должна быть указана поверхность, которая будет направляющей в следующей команде движения.

Операторы НАПРТ *и* НАПРВ являются вспомогательными и используются совместно с оператором ИДИ в следующих случаях:

1) если при использовании оператора ИДИ с одной поверхностью (кроме поверхности детали) надо осуществить подход не к ближайшей точке поверхности;

2) если при использовании оператора ИДИ с двумя поверхностями (не считая поверхности детали) необходимо выбрать одну из двух возможных точек пересечения *НП* и *ОП*;

3) если инструмент уже находится в требуемом положении относительно *НП*;

4) если в операторе ИДИ в качестве *НП* указан объект типа КОНТУР.

В первом случае отсутствие оператора, задающего направление подхода, вызовет подход к ближайшей точке поверхности, а в остальных трех случаях приведет к ошибке.

Операторы НАПРТ и НАПРВ не управляют перемещением по третьей координате, т. е. направление подхода к *НП* определяется проекцией вектора подхода на горизонтальную плоскость.

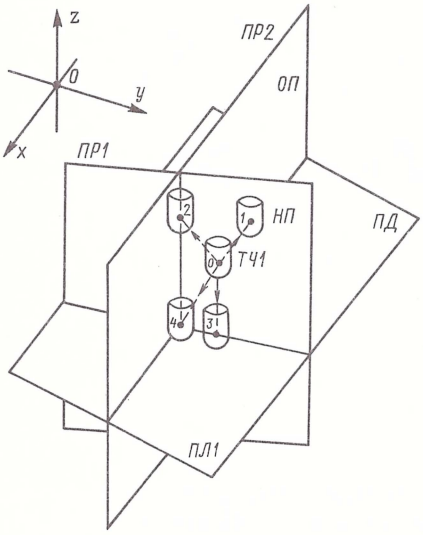


Рис. 2.78

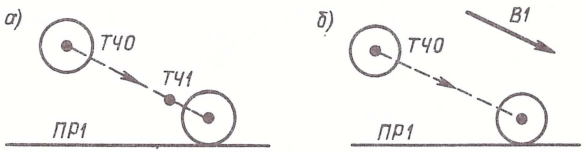


Рис. 2.79

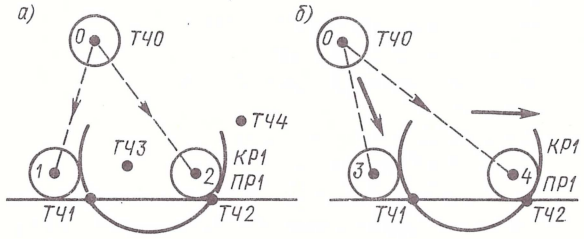


Рис. 2.80

Формат оператора НАПРТ:

НАПРТ $\left\{\begin{matrix} точка \\ x, y \end{matrix}\right\}$

где *точка* — имя точки; *x*, *y* — переменные или арифметические выражения, значения которых являются соответственно координатами *x* и *y* точки.

Формат оператора НАПРВ:

НАПРВ $\left\{\begin{matrix} вектор \\ x, y \end{matrix}\right\}$

где *вектор* — имя вектора; *x*, *y* — переменные или арифметические выражения, значения которых являются соответственно проекциями *x* и *y* вектора.

На рис. 2.79, *а*, *б* представлены методы подхода к НП в направлении точки и вектора соответственно:

(*а*) ИЗ ТЧО
НАПРТ ТЧ1
ИДИ ДО ПР1

(*б*) ИЗ ТЧО
НАПРВ В1
ИДИ ДО ПР1

В случае, когда *НП* и *ОП* пересекаются в двух точках, целевая точка подхода выбирается с помощью операторов НАПРТ или НАПРВ. Если используется НАПРТ, то целевой выбирается точка, ближайшая к точке, заданной в этом операторе. Если используется НАПРВ, то указанный там вектор используется для выбора из двух точек так же как и модификаторы типа ХБ — УМ в геометрических определениях (рис. 2.80), например:

перемещение *0—1* (рис. 2.80, *а*) — оператор ИДИ задает переход в точку *ТЧ1*, так как она ближе к *ТЧ3*, чем *ТЧ2*:

ИЗ ТЧ0
НАПРТ ТЧ3
ИДИ ДО ПР1 ДО КР1

перемещение *0—2* (рис. 2.80, *а*) — инструмент переходит в точку *ТЧ2*, так как она ближе к *ТЧ4*, чем *ТЧ1*

ИЗ ТЧ0
НАПРТ ТЧ4
ИДИ ДО ПР1 ДО КР1

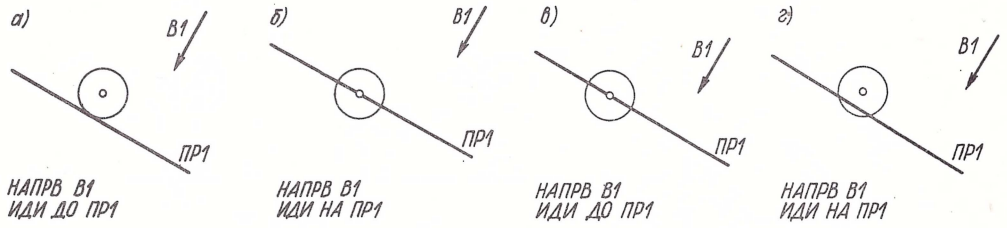


Рис. 2.81

перемещение *0—3* (рис. 2.80, *б*):

ИЗ ТЧ0
НАПРВ ВЕКТОР (ТЧ0, ТЧ1)
ИДИ ДО ПР1 ДО КР1

перемещение *0—4* (рис. 2.80, *б*):

ИЗ ТЧ0
НАПРВ ВЕКТОР (1, 0, 0)
ИДИ ДО ПР1 ДО КР1

Если инструмент уже подведен к *НП*, то для вывода инструмента в требуемое положение относительно *НП* или для задания направления, относительно которого будет выбран оператор непрерывного движения, следует указать направление подхода операторами НАПРВ или НАПРТ (рис. 2.81, *а*, *б*). При этом запрещен вывод инструмента в направлении, встречном указанному в операторах НАПРВ или НАПРТ (рис. 2.81, *в*). В исходном положении инструмент может касаться *НП* или его центр может находиться на *НП*; вариант, показанный на рис. 2.81, *г*, недопустим.

Если в операторе ИДИ в качестве *НП* указан объект типа КОНТУР, то следует указать направление подхода (см. п. 2.11):

ИЗ ТЧ1
НАПРВ В1
ИДИ ДО КОНТ1

Оператор ПОВДЕТ предназначен для управления поверхностью детали при поточечном и непрерывном движении.

Оператор имеет следующий формат:

$$\text{ПОВДЕТ} \left\{ \begin{matrix} \textit{число} \\ \textit{имя} \\ \text{ОТМЕН} \end{matrix} \right\}$$

число — число или арифметическое выражение, задающее *ПД* как плоскость, параллельную плоскости *xOy* и отстоящую от нее на указанном расстоянии; *имя* — символическое имя плоскости, которая будет использоваться как *ПД* в последующих командах движения; ОТМЕН — служебное слово, вызывающее отмену ранее введенной *ПД* и возврат к плоскости *xOy*.

При поточечном движении координата *z* каждой точки будет вычисляться в соответствии с заданной *ПД*. В отличие от оператора ПЛЗ, который задает координату *z* при построении точек, оператор ПОВДЕТ используется с командами движения.

При непрерывном движении применение оператора ПОВДЕТ равносильно заданию в операторе ИДИ поверхности детали с модификатором НА. Вводя новые операторы ПОВДЕТ, можно переходить от одной поверхности детали к другой, не прерывая непрерывного движения. Применять одновременно операторы ПОВДЕТ и *ПД* в операторе ИДИ нельзя.

Независимо от того, как в программе определена поверхность детали (в операторе ИДИ или в операторе ПОВДЕТ), инструмент будет находиться в контакте с ней до тех пор, пока ПД не будет отменена командой ПОВДЕТ ОТМЕН.

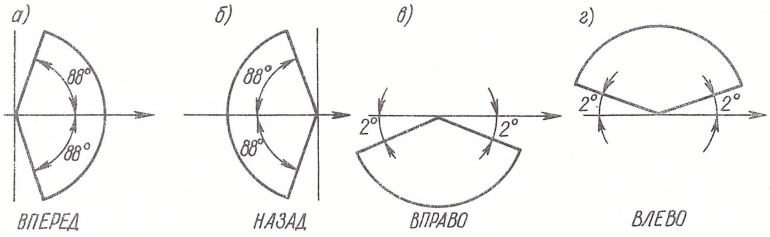


Рис. 2.82

После выхода на направляющую поверхность движение по каждому из геометрических элементов контура описывается в следующем формате:

$$\left\{\begin{matrix}\text{ВПЕРЕД}\\ \text{НАЗАД}\\ \text{ВЛЕВО}\\ \text{ВПРАВО}\end{matrix}\right\}\ \text{НП}\ \left\{\begin{matrix}\text{ДО}\\ \text{НА}\\ \text{ЗА}\\ \text{КАС}\end{matrix}\right\}\ [\textit{число}\ [\text{ПЕРЕСЕЧ}]]\ \text{ОП}$$

ВПЕРЕД, НАЗАД, ВПРАВО, ВЛЕВО — служебные слова, указывающие направление перемещения по направляющей поверхности; *НП* и *ОП* — направляющая и ограничивающая поверхности; *число* — номер пересечения *НП* с *ОП*, до которого задано движение по *НП*; КАС — модификатор, определяющий движение по *НП* до точки касания с *ОП*; ДО, НА, ЗА — модификаторы, определяющие положение инструмента относительно *ОП* (аналогично модификаторам оператора ИДИ).

Направление движения выбирается относительно предыдущего перемещения в соответствии с рис. 2.82, *а—г* (секторами ограничены области действия служебных слов, указывающих направление перемещения).

При движении по окружности направлением движения считается направление касательной к окружности в точке пересечения *НП* и *ОП*. Поэтому в случае касания *НП* и *ОП* следует применять только операторы ВПЕРЕД и НАЗАД (рис. 2.83, *а*, *б*).

Пример программирования непрерывного движения для контура, изображенного на рис. 2.84 (из положения *1* в положение *5*):

ВПЕРЕД П1 ЗА КР1
ВЛЕВО КР1 КАС К2
ВПЕРЕД КР2 ДО ПР2
ВПРАВО ПР2 НА П1

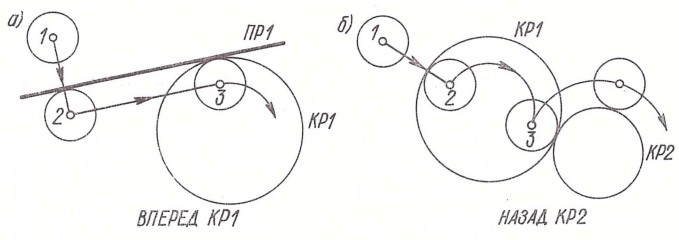


Рис. 2.83

Поверхности, указанные в операторе непрерывного движения, могут пересекаться и не один раз. Если не указывать номер пересечения, то инструмент будет выведен на (до, за) первое пересечение НП и ОП. Поэтому, когда требуется остановиться на последующих пересечениях, следует указывать номер пересечения НП с ОП.

Номер пересечения отсчитывается для каждого оператора движения в отдельности, вне связи с предыдущими операторами непрерывного движения. Причем, если перед

выполнением оператора движения инструмент уже касался НП и ОП, это пересечение считается первым.

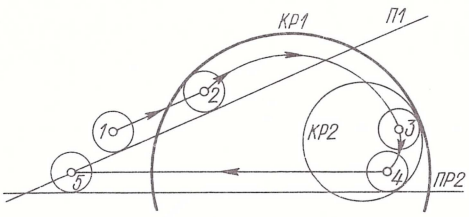


Рис. 2.84

Примеры программирования непрерывного движения:

для контура, изображенного на рис. 2.85, *а*,—

1—2 ВПЕРЕД ПР1 ЗА КР1
или ВПЕРЕД ПР1 ЗА 1 КР1
2—3 ВПЕРЕД ПР1 ЗА 2 КР1

для контура, изображенного на рис. 2.85, *б*,—

1—2 ВПЕРЕД ПР2 ДО 2 КР2
2—3 ВЛЕВО КР2 НА 2 ПР2
3—4 ВПЕРЕД КР2 НА 2 ПР2

Номер пересечения, указанный в операторе движения, не должен превышать общего числа пересечений данных НП и ОП, в противном случае ситуация считается ошибочной (например, задание второго пересечения для двух прямых). Исключение составляет третье пересечение окружности с прямой, под которым подразумевается полный обход окружности.

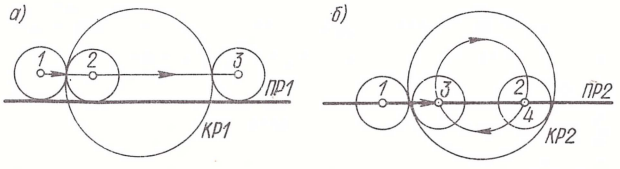


Рис. 2.85

Примеры программирования с «третьим» пересечением:

для контура, изображенного на рис. 2.86, *а*,—

ВПЕРЕД ПР1 ДО КР2
НАЗАД КР2 ДО 3 ПЕРЕСЕЧ ПР1

для контура, изображенного на рис. 2.86, *б*,—

ВПЕРЕД ПР2 ЗА КР3
ВЛЕВО КР3 ДО 3 ПР2

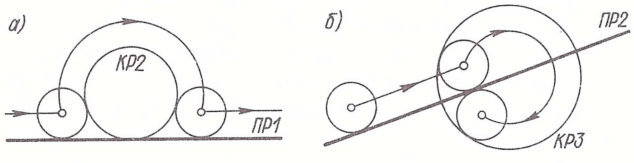


Рис. 2.86

Если поверхность детали является наклонной плоскостью, то команды движения следует задавать так же, как и в случае расположения обрабатываемого контура в плоскости *xOy*. Истинное перемещение инструмента представляет собой результат проектирования на ПД в направлении оси *z* эквидистанты к этому контуру (с учетом формы инструмента и его положения относительно ПД).

На рис. 2.87 показано программируемое и истинное перемещения инструмента при заданной поверхности детали для следующей программы:

```
МЕТОД ЛИНЕЙН
ИЗ ТЧ1
ИДИ ДО ПР1 ДО ПЛ1 ЗА КР1
ВЛЕВО ПР1 ДО ПР2
ВЛЕВО ПР2 ДО КР1
ВЛЕВО КР1 ДО ПР1
ПОВДЕТ ОТМЕН
ВТОЧКУ ТЧ1
```

Неявное определение ограничивающей поверхности имеет место в том случае, если известно положение инструмента относительно контура обрабатываемой детали и ограничивающая поверхность и относящийся к ней модификатор в промежуточном операторе движения могут быть опущены. Они выбираются системой по направляющей поверхности следующего оператора движения. При этом необходимо задать положение инструмента относительно контура при помощи одного из операторов:

ИНСТНА — инструмент на контуре детали;

ИНСТЛВ — инструмент слева от контура (по направлению движения);

ИНСТПР — инструмент справа от контура (по направлению движения).

Несколько следующих подряд операторов движения одного типа могут заменяться одним, при этом операнды перечисляются через двоеточие. Число пересечений и

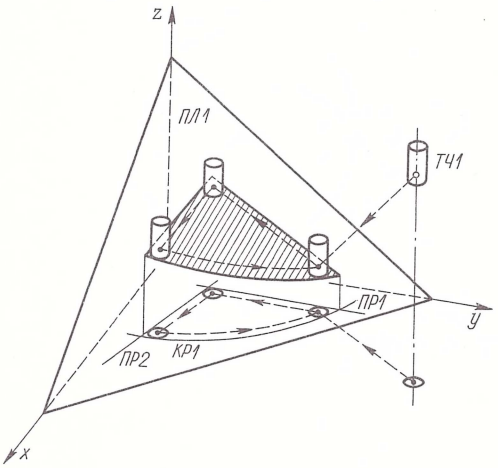


Рис. 2.87

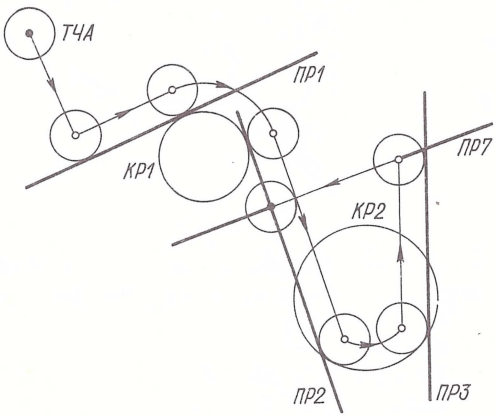


Рис. 2.88

ограничивающая поверхность с соответствующим модификатором задаются как обычно. При необходимости можно переходить от неявной ограничивающей поверхности к явной и наоборот.

Данный прием программирования позволяет значительно уменьшить объем исходной информации, однако при этом затрудняется понимание диагностических сообщений, связанных с номером строки.

На рис. 2.88 изображен контур, при программировании которого целесообразно использовать операторы непрерывного движения с неявной ограничивающей поверхностью:

```
ИЗ ТЧА
ИДИ ДО ПР1
ИНСТЛВ
ВЛЕВО ПР1
ВПЕРЕД КР1 : ПР2 ДО 2 ПЕРЕСЕЧ
ВЛЕВО КР2 : ПР3 НА ПР7
ВЛЕВО ПР7 НА ПР2
```

Последний оператор движения всегда должен содержать явно заданную ограничивающую поверхность. После операторов движения, не содержащих ограничивающей поверхности, запрещено использовать операторы постпроцессора и управления интерполяцией.

Совместное применение операторов поточечного и непрерывного движений имеет некоторые особенности. Поточечное движение прерывает непрерывное движение, поэтому после операторов ПРИРАЩ и ВТОЧКУ невозможно продолжить непрерывное движение при помощи операторов ВПРАВО, ВЛЕВО, ВПЕРЕД, НАЗАД. Для возобновления непрерывного движения следует задать оператор ИДИ.

Исключение составляет перемещение по третьей координате при незаданной поверхности детали (рис. 2.89):

ВПЕРЕД ПР1 ДО КР2
ПРИРАЩ 50
ВПРАВО КР2

Фиктивное движение, т. е. расчет перемещений инструмента без передачи их в управляющую программу можно запрограммировать с помощью оператора НЕОБР и отменяющего его оператора ОБРАБ. Этот прием используют для того, чтобы переместить инструмент в положение, которое описать явным образом неудобно или

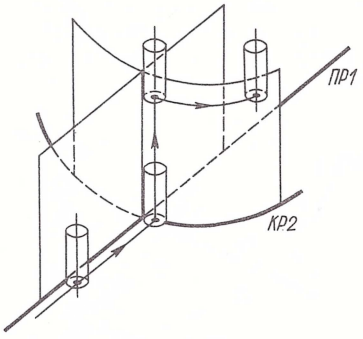


Рис. 2.89

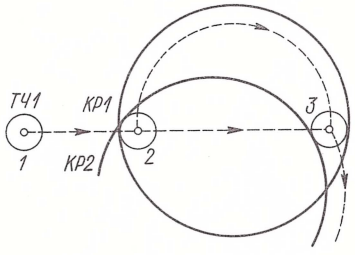


Рис. 2.90

невозможно. Например, для случая, изображенного на рис. 2.90, непосредственный переход из положения *1* в положение *3* программируется операторами:

ИЗ ТЧ1
НЕОБР
ИДИ ЗА КР2 ЗА КР1
ВЛЕВО КР1 ДО 2 КР2
ОБРАБ
ВПЕРЕД КР2 ....

При этом исключается движение по окружности *КР1* из положения *2* в положение *3*.

2.10.5. Преобразование траектории движения инструмента

При программировании сложных деталей бывает удобно кодировать элементы контура не в базовой системе координат детали, а в некоторых местных системах координат. При этом необходимо задать положение каждой местной системы координат в базовой системе координат детали, для чего используются матричные преобразования.

Возможны два способа использования матриц: в геометрических определениях и в командах движения.

В первом случае новый геометрический объект определяется посредством матричного преобразования, производимого над ранее определенным геометрическим объектом. Во втором случае преобразованию подвергается отмеченный участок траектории, для чего используют операторы ТРАНС и ТРАНСРЕЗ.

Оператор ТРАНС предназначен для преобразования направляющих и ограничивающих поверхностей, относительно которых описывается движение. Оператор ТРАНСРЕЗ используют для преобразования точек рассчитанной траектории.

Оператор ТРАНС имеет следующий формат:

ТРАНС $\left\{\begin{matrix}\textit{матрица}\\ \text{ОТМЕН}\end{matrix}\right\}$

где *матрица* — символическое имя матрицы.

Матрица, заданная в этом операторе, должна быть плоской, т. е. это либо матрица поворота в плоскости xOy, либо матрица переноса по координатам x и y, либо матрица симметрии относительно прямой, либо произведение матриц такого вида.

Оператор ТРАНС выполняет преобразование траектории движения инструмента в соответствии с заданной матрицей до применения нового оператора ТРАНС. Оператор ТРАНС ОТМЕН задает возврат к исходной системе координат. Если в программе последовательно применяются несколько операторов ТРАНС, то следует учесть, что каждый следующий оператор определяет преобразование исходной системы координат, а не текущей; оператор ТРАНС ОТМЕН задает возврат к исходной системе координат, а не к действовавшей ранее.

Например, необходимо обработать три одинаковых элемента (*1—3*), расположенных так, как показано на рис. 2.91. Фрагмент программы на языке Техтран имеет вид:

```
. . .
команды обработки элемента 1
МАТР М1 = ПОВОРОТ, 30
ТРАНС М1
команды обработки элемента 1
МАТР М2 = ПОВОРОТ, 60
ТРАНС М2
команды обработки элемента 1
ТРАНС ОТМЕН
. . .
```

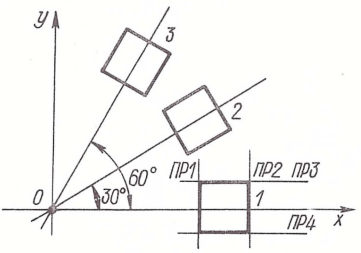


Рис. 2.91

Оператор ТРАНС действует на поверхности движения (прямые, окружности и др.), а не на получаемые точки траектории. Поэтому следует иметь в виду, что неправильное употребление этого оператора в середине непрерывного движения может привести к разрыву траектории (рис. 2.92).

Предполагается вывести инструмент последовательно из Т40 в позиции *1—4*:

```
ИЗ Т40
ИДИ ДО КР1
ВЛЕВО КР1 ДО ПР1
ТРАНС МАТР (ПОВОРОТ, 30)
ВПРАВО ПР1 ДО КР2
```

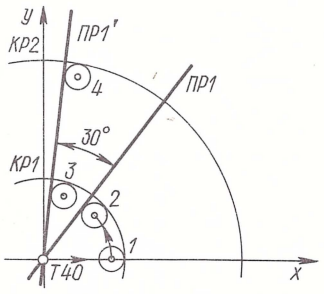


Рис. 2.92

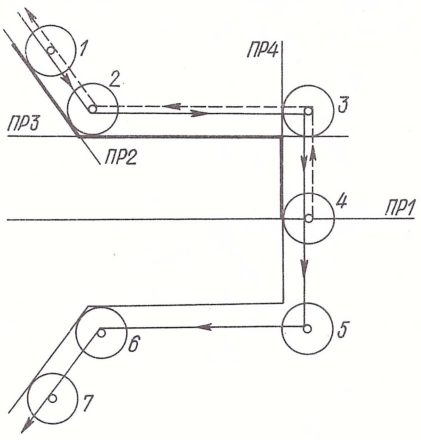


Рис. 2.93

После выполнения оператора 3 инструмент оказался в позиции *2*. После выполнения оператора 4 прямая *ПР1* заняла положение *ПР1'*, инструмент же остался в позиции *2*. В связи с этим выполнить оператор 5 невозможно.

При использовании матрицы симметрии относительно прямой операторы движения, следующие после оператора ТРАНС, должны описывать движение по исходному контуру в обратном направлении (кроме первого перемещения после оператора ТРАНС). При этом не требуется изменять положение инструмента относительно контура, если оно было задано оператором ИНСТЛВ или ИНСТПР.

Движение по контуру, изображенному на рис. 2.93 (– – → — траектория, описанная после оператора ТРАНС; ⟶ — фактическая траектория движения инструмента), программируется следующим образом (табл. 2.4).

Оператор ТРАНСРЕЗ имеет следующий формат:

ТРАНСРЕЗ $\left\{\begin{matrix} имя \\ \text{ОТМЕН} \end{matrix}\right\}$

где *имя* — символическое имя матрицы.

Матрица, заданная в этом операторе, может быть любой (в том числе и трехмерной). В отличие от оператора ТРАНС оператор ТРАНСРЕЗ осуществляет преобразование по заданной матрице рассчитанных точек траектории. Действие его распространяется до нового оператора ТРАНСРЕЗ или до оператора ТРАНСРЕЗ ОТМЕН.

Таблица 2.4

Оператор программы	Участок траектории	
	запрограммированной	фактической
ИНСТЛВ		
ВПЕРЕД ПР2	....—*1*—*2*	....—*1*—*2*
ВЛЕВО ПР3	*2*—*3*	*2*—*3*
ВПРАВО ПР4	*3*—*4*	*3*—*4*
НА ПР1		
МАТР М1=ПР1		
ТРАНС М1		
ВПЕРЕД ПР4	*4*—*3*	*4*—*5*
ВЛЕВО ПР3	*3*—*2*	*5*—*6*
ВПРАВО ПР2	*2*—*1*—....	*6*—*7*—....

Различие в результатах, полученных при использовании операторов ТРАНС и ТРАНСРЕЗ в приведенном примере, иллюстрируется на рис. 2.94 (– – – → — программируемое перемещение; ⟶ — фактическое перемещение). Например:

```
ТЧ0=0,0
ТЧ1=—100,0
ПР1=ТЧ0, 60
МАТР М1=ПОВХУ, —90
ИЗ ТЧ1
НАПРТ ТЧ0
ИДИ НА ПР1
ТРАНС М1
ВПРАВО ПР1 НА ОКРУЖН (ТЧ0, 50)
```

Оператор ТРАНС выполняет преобразование поверхностей. Поэтому после применения оператора ТРАНС прямая *ПР1* примет положение *ПР1′*, а оператор ВПРАВО вызовет перемещение инструмента из *ТЧ0* в точку *1* (см. рис. 2.94, *а*).

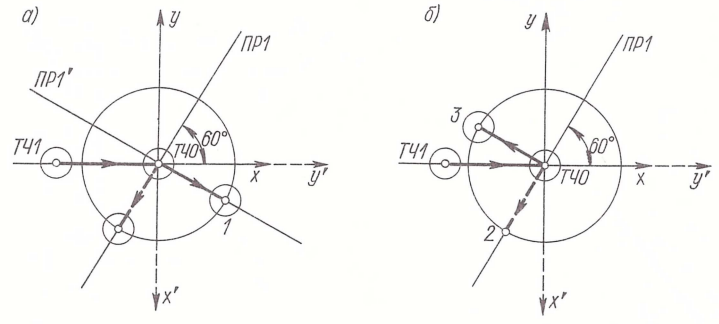


Рис. 2.94

Если в этой же программе заменить оператор ТРАНС М1 оператором ТРАНСРЕЗ М1, то преобразованию будет подвергнута точка траектории *2*, которая займет положение *3*, куда и переместится инструмент из *ТЧ0* (см. рис. 2.94, *б*).

2.11. Контуры

Наряду с элементарными типами геометрических объектов в языке Техтран имеется комбинированный тип — КОНТУР. Геометрические объекты этого типа являются совокупностями отрезков прямых и дуг окружностей. Они могут использоваться в командах непрерывного движения аналогично элементарным геометрическим объектам, т. е. инструмент может быть выведен на контур оператором ИДИ, а движение по нему осуществляется с помощью операторов ВЛЕВО, ВПРАВО, ВПЕРЕД, НАЗАД.

Различают простые и составные контуры. Простой контур — комбинация элементарных геометрических объектов, а составной контур является комбинацией нескольких простых контуров.

Простой контур. Описание простого контура состоит из последовательности операторов непрерывного движения, ограниченных заголовком контура и служебным словом КОНКОН:

```
[КОНТУР] имя=ИЗ точка, модификатор
    оператор1
    оператор2
    ...
    оператор n-1
    оператор n
КОНКОН
```

где КОНТУР — служебное слово, задающее тип определяемого объекта (если он еще не известен); *имя* — имя, присвоенное контуру; *точка* — имя первой точки контура (если контур замкнутый, то любой точки контура); *модификатор* — любой из модификаторов ХБ—ХМ—УБ—УМ, задающий направление, относительно которого выбирается оператор$_1$; *оператор*$_1$, *оператор*$_2$, ..., *оператор*$_n$ — операторы движения.

Простой контур называется замкнутым, если начальная и конечная точки контура совпадают, и незамкнутым, если точки не совпадают. Направление движения в *операторе*$_1$ определяется относительно модификатора, указанного в заголовке контура. Проще всего в качестве *оператора*$_1$ употреблять оператор ВПЕРЕД. Направление движения в последующих операторах указывается относительно предыдущего направления движения по правилам, изложенным в п. 2.10.

Оператор$_i$ $(1 \leqslant i \leqslant n)$ может быть оператором с двумя поверхностями

$$\left\{\begin{matrix}\text{ВПЕРЕД}\\ \text{НАЗАД}\\ \text{ВЛЕВО}\\ \text{ВПРАВО}\end{matrix}\right\} \text{НП} \left\{\begin{matrix}\text{ДО}\\ \text{НА}\\ \text{ЗА}\\ \text{КАС}\end{matrix}\right\} [\textit{число}\ [\text{ПЕРЕСЕЧ}]]\ \text{ОП}$$

либо оператором с одной поверхностью

$$\left\{\begin{matrix}\text{ВПЕРЕД}\\ \text{НАЗАД}\\ \text{ВЛЕВО}\\ \text{ВПРАВО}\end{matrix}\right\} \text{НП}\ [\textit{число}\ \text{ПЕРЕСЕЧ}]$$

В описании замкнутого контура направляющие поверхности в первом и последнем операторах могут совпадать. В описании контуров кроме операторов движения можно использовать операторы ПОВТОР и ЕСЛИ (см. п. 2.12).

Незамкнутый контур, изображенный на рис. 2.95, описывается таким образом:

```
КОНТУР КОН1=ИЗ ТЧ1, ХБ
    ВПЕРЕД ПР1
    ВЛЕВО КР1 2 ПЕРЕСЕЧ
    ВЛЕВО ПР1 НА ПР2
КОНКОН
```

Применим в описании контура (рис. 2.95) условный оператор для выбора обхода верхней или нижней полуокружности в зависимости от знака переменной *P*:

```
КОНТУР КОН1=ИЗ ТЧ1, ХБ
    ВПЕРЕД ПР1
    ЕСЛИ(P>0)ТО
        ВЛЕВО КР1 2 ПЕРЕСЕЧ
        ВЛЕВО ПР1 НА ПР2
    ИНАЧЕ
        ВПРАВО КР1 2 ПЕРЕСЕЧ
        ВПРАВО ПР1 НА ПР2
    КОНЕСЛИ
КОНКОН
```

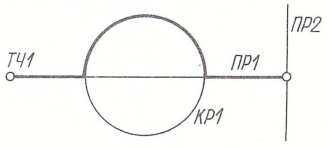


Рис. 2.95

Замкнутый контур, изображенный на рис. 2.96, может быть задан двумя способами. Обычная форма записи следующая:

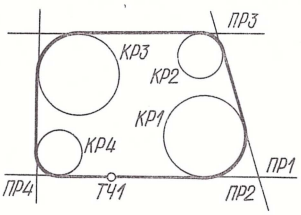


Рис. 2.96

```
КОНТУР КОН2=ИЗ ТЧ1, ХБ
    ВПЕРЕД ПР1
    ВПЕРЕД КР1
    ВПЕРЕД ПР2
    ВПЕРЕД КР2
    ВПЕРЕД ПР3
    ВПЕРЕД КР3
    ВПЕРЕД ПР4
    ВПЕРЕД КР4
    ВПЕРЕД ПР1
КОНКОН
```

Сокращенная форма записи имеет вид:

```
КОНТУР КОН2=ИЗ ТЧ1, ХБ
    ВПЕРЕД ПР1:КР1:ПР2:КР2:ПР3:КР3:ПР4:КР4:ПР1
КОНКОН
```

Кроме операторов движения в определении контура можно использовать и операторы постпроцессора. При этом следует соблюдать осторожность, так как если направление обхода контура не соответствует направлению, выбранному при описании, то оператор постпроцессора, записанный перед некоторым сегментом контура, будет в действительности относиться к предыдущему сегменту.

Составной контур. Составной контур представляет собой комбинацию ранее определенных контуров. Если *контур*$_1$, *контур*$_2$, *контур*$_3$, *контур*$_n$ — ранее определенные контуры, *матрица*$_1$, *матрица*$_2$, *матрица*$_3$, *матрица*$_n$ — плоские матрицы, а *число*$_1$, *число*$_2$, *число*$_3$, *число*$_n$ — числа, переменные или арифметические выражения, то оператор вида

КОНТУР. *имя*=*число*$_1$, *матрица*$_1$, *контур*$_1$, %
, *число*$_2$, *матрица*$_2$, *контур*$_2$, %
, *число*$_3$, *матрица*$_3$, *контур*$_3$, %
. . . . . .
, *число*$_n$, *матрица*$_n$, *контур*$_n$

определяет новый контур, который строится следующим образом. Каждый i-й контур преобразуется по заданной матрице и присоединяется к предыдущему контуру начальной точкой, если *число*$_i$ положительное, или конечной точкой, если *число*$_i$ отрицательное. Если же *число*$_i$ равно нулю, то контур пропускается. По умолчанию предполагается, что *число*$_i$=1. Начальной точкой результирующего контура является начальная точка *контура*$_1$, если *число*$_1$ положительное, и конечная его точка, если *число*$_1$ отрицательное. *Матрица*$_1$ — любая плоская матрица, *матрица*$_i$ ($i \geqslant 2$) — матрица поворота или симметрии; сдвиг, если он есть, не учитывается. Служебное слово КОНТУР должно присутствовать в том случае, когда определение типа не было сделано ранее.

Если после знака «=» в определении составного контура нет ничего, то это означает, что контур имеет многострочный формат с ограничителем КОНКОН, где элементы составного контура отделены друг от друга концами строк или символами «;».

Например, контур *К* (рис. 2.97, *в*), состоящий из контуров *К1* и *К2* (рис. 2.97, *а*, *б*), описывается следующим образом:

```
. . . . . .
МАТР М1=ПОВОРОТ, 45
КОНТУР К1=ИЗ ТЧ1, . . . . . .
. . . . . .
КОНКОН
КОНТУР К2=ИЗ ТЧ2, . . . . . .
. . . . . .
КОНКОН
КОНТУР К=М1, К1, К2
```

Еще один пример построения составного контура приведен на рис. 2.98, *б* (на рис. 2.98, *а* — исходный контур):

М1 = ПОВОРОТ, −90
КОНТУР К = К1, М1, К1, −1, К1

На рис. 2.99, *б* приведен пример составного контура, который получается симметричным преобразованием исходного контура (на рис. 2.99, *а* — исходный контур):

КОНТУР К, К1
М1 = ПРХ
М2 = ПРУ
М3 = М1, М2
К1 = ...
К = +1, К1, −1, М1, К1, +1, М3, К1, −1, М2, К1
(или К = К1, −1, М1, К1, М3, К1, −1, М2, К1)

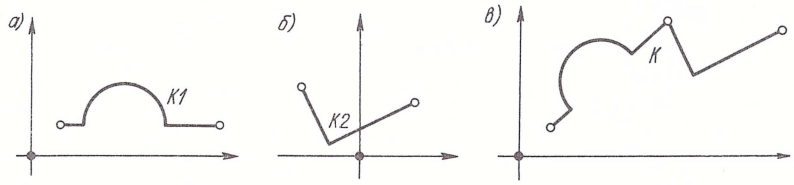


Рис. 2.97

В определениях составных контуров можно использовать операторы ЕСЛИ и ПОВТОР. Применим оператор цикла при построении контура *К* (рис. 2.100, *б*), получающегося многократным повторением контура *К1* (рис. 2.100, *а*):

```
КОНТУР К=
    ПОВТОР И=1,5
        МАТР (ПОВОРОТ, 72*(И −1)), К1
    КОНЦИКЛ
КОНКОН
```

Частным случаем описания составного контура является оператор присваивания: А = Б или КОНТУР А = Б, где А и Б — объекты типа КОНТУР.

Определение табулированной кривой. Табулированная кривая (СПЛАЙН, таблично заданная кривая) — это гладкая кривая, интерполирующая заданный набор точек и состоящая из отрезков прямых и дуг окружностей. Табулированная кривая имеет направление от первой заданной точки к последней (т. е. при построении кривой от точки $ТЧ_1$ к точке $ТЧ_n$ и от точки $ТЧ_n$ к точке $ТЧ_1$ результаты совпадать не будут).

Табулированные кривые имеют тип КОНТУР и с ними можно осуществлять любые операции, имеющие смысл для незамкнутых контуров. Так, например, они могут быть компонентами составного контура. Их геометрическое определение в языке Техтран имеет вид:

КОНТУР *имя* = СПЛАЙН [, *прямая*$_1$], *точка*$_1$, *точка*$_2$, . . . , *точка*$_n$, [*прямая*$_2$]

где КОНТУР — служебное слово, задающее тип определяемого объекта, если он не был объявлен ранее; СПЛАЙН — служебное слово, задающее вид определяемого контура; *точка*$_1$, *точка*$_2$, . . . , *точка*$_n$ — опорные точки кривой (число точек не ограничено); *прямая*$_1$, *прямая*$_2$ — имена необязательных прямых, касательных к кривой в начальной и конечной точках.

Как и в определении контура, в определении табулированной кривой можно использовать операторы ПОВТОР и ЕСЛИ. Определение с оператором цикла имеет вид:

```
[КОНТУР] имя=СПЛАЙН, прямая1
    точка1, точка2, . . . , точкаk
    ПОВТОР счетчик=нач, кон, шаг
        массив точек (счетчик)
    КОНЦИКЛ
    точкаk+1, точкаk+2, . . . , точкаm
    прямая2
КОНКОН
```

Например:

```
КОНТУР Т1=СПЛАЙН, ПР1
    ПОВТОР И=1,50
        ТЧ(И)
    КОНЦИКЛ
    ПР2
КОНКОН
```

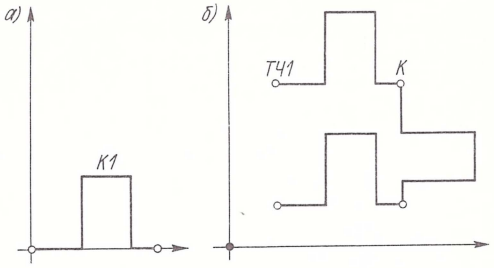


Рис. 2,98

Определение табулированной кривой с условным оператором имеет вид:

```
[КОНТУР] имя=СПЛАЙН [, прямая1]
    точка1, точка2, . . . , точкаk
    ЕСЛИ (логическое выражение) ТО
     точкаk+1, точкаk+2, . . . , точкаl
    ИНАЧЕ
     точкаl+1, точкаl+2, . . . , точкаm
    КОНЕСЛИ
     точкаm+1, точкаm+2, . . . , точкаn
КОНКОН
```

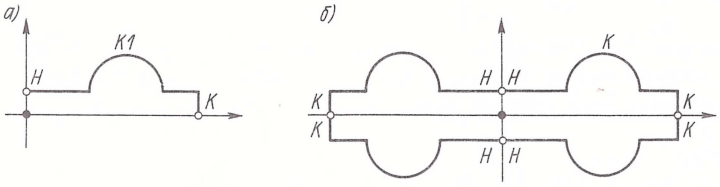


Рис. 2.99

Например:

```
КОНТУР Т2=СПЛАЙН, ПР1
    ЕСЛИ (А>=В) ТО
    ТЧ1, ТЧ2, ТЧ3
    ИНАЧЕ
    ТЧ4, ТЧ5, ТЧ6
    КОНЕСЛИ
    ПР2
КОНКОН
```

Оператор цикла и условный оператор могут встречаться в определении табулированной кривой в различных комбинациях:

```
КОНТУР Т3=СПЛАЙН, ПР1
    ТЧ1, ТЧ2, ТЧ3
    ЕСЛИ (А>=В) ТО
        ПОВТОР И=1, К
            ТЧ(И)
        КОНЦИКЛ
    ИНАЧЕ
        ПОВТОР И=1, М
            ЕСЛИ (Л=1) ТО
                ТК(И)
            ИНАЧЕ
                ТК(М—И)
            КОНЕСЛИ
        КОНЦИКЛ
    КОНЕСЛИ
    ТЧ4, ТЧ5, ТЧ6
    ПР2
КОНКОН
```

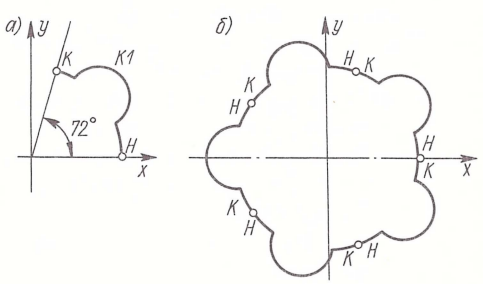


Рис. 2.100

Подход к контуру. Существует два способа подхода к контуру: при помощи оператора ИДИ и при помощи операторов непрерывного движения ВПЕРЕД, НАЗАД, ВЛЕВО, ВПРАВО.

При использовании оператора ИДИ следует указывать направление подхода к контуру командой НАПРВ или НАПРТ (рис. 2.101):

```
ИЗ ТЧ1
НАПРВ В1
ИДИ ЗА К1
```

При использовании операторов непрерывного движения инструмент следует вывести на контур с помощью любой команды непрерывного движения с двумя поверхностями. Число пересечений не должно превышать восьми. Если поверхность подхода совпадает с одним из сегментов контура, то инструмент выводится в первую или последнюю точку сегмента.

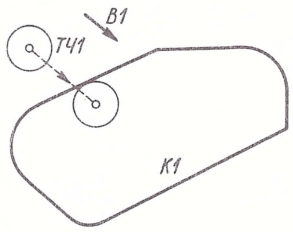


Рис. 2.101

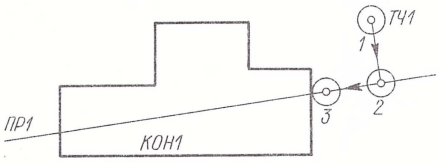


Рис. 2.102

На рис. 2.102 приведен пример подхода к контуру по прямой, заданный следующими командами движения:

```
ИЗ ТЧ1
ИДИ НА ПР1
ВПРАВО ПР1 ДО КОН1
```

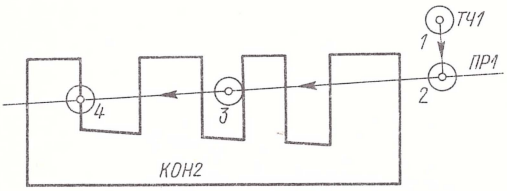


Рис. 2.103

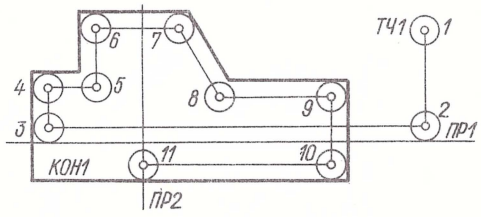


Рис. 2.104

В случае многократных пересечений (рис. 2.103) вывод инструмента из позиции *1* в позицию *3* задается следующим образом:

```
ИЗ ТЧ1
ИДИ НА ПР1
ВПРАВО ПР1 ЗА 4 ПЕРЕСЕЧ КОН2
```

Перемещение инструмента из позиции *1* в позицию *4* программируется операторами:

```
ИЗ ТЧ1
ИДИ НА ПР1
ВПРАВО ПР1 НА 7 ПЕРЕСЕЧ КОН2
```

Движение по контуру. Для программирования движения инструмента по контуру можно задать любой из операторов ВПЕРЕД, НАЗАД, ВЛЕВО, ВПРАВО с двумя поверхностями, указав требуемое число пересечений. Ограничивающей поверхностью может быть только прямая или окружность (но не контур и не табулированная кривая).

Например, для контура, показанного на рис. 2.104, перемещение инструмента из позиции *1* в позиции *2—11* программируется следующим образом:

ИЗ ТЧ1
ИДИ ДО ПР1
ВПРАВО ПР1 ДО 2 ПЕРЕСЕЧ КОН1
ВПРАВО КОН1 НА 2 ПЕРЕСЕЧ ПР2

Аналогично можно запрограммировать и перемещение инструмента по незамкнутому контуру (рис. 2.105):

ИЗ ТЧ1
НАПРТ ТЧА
ИДИ ДО К1
ВПРАВО К1 НА ПР1

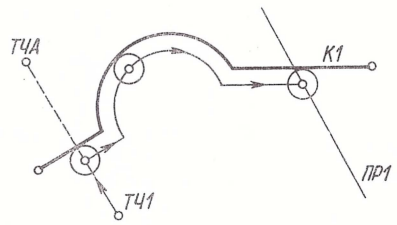


Рис. 2.105

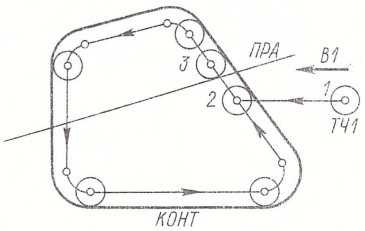


Рис. 2.106

Полный обход всего контура программируется оператором непрерывного движения с одной поверхностью, которой является контур:

$$\left\{\begin{matrix}\text{ВПЕРЕД}\\ \text{НАЗАД}\\ \text{ВЛЕВО}\\ \text{ВПРАВО}\end{matrix}\right\}\ \textit{имя}$$

где *имя* — символическое имя контура.

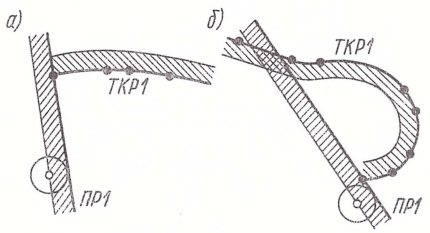


Рис. 2.107

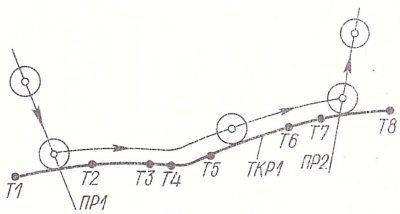


Рис. 2.108

Совершив полный обход контура, инструмент возвращается в положение, в котором он находился перед использованием оператора этого вида.

Например, полный обход контура представленного на рис. 2.106, задается операторами:

ИЗ ТЧ1
НАПРВ В1
ИДИ ЗА КОНТ # поз *1—2*
ВПРАВО КОНТ ЗА ПРА # поз. *2—3*
ВПЕРЕД КОНТ # поз. *3—2—3*

Оператор обхода контура с одной поверхностью допускается употреблять только для замкнутого контура.

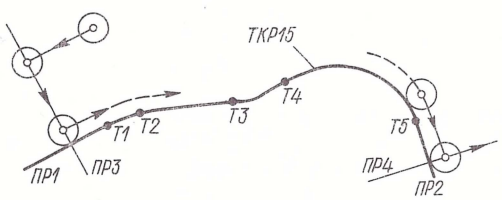


Рис. 2.109

Особенности движения по табулированным кривым. Поскольку табулированная кривая является разновидностью контура, то все сказанное выше о командах движения для контуров применимо и к табулированным кривым. Необходимо помнить, что при непрерывном движении по контуру не только сами поверхности, но и их эквидистанты должны иметь требуемое число пересечений. Эквидистанта контура (и табулированной кривой) ограничена нормалями в начальной и конечной точках контура. Например, в случаях, изображенных на рис. 2.107, *а*, *б*, невозможен выход в начало табулированной кривой, так как эквидистанты к *ПР1* и *ТКР1* не пересекаются в начальной точке. Примеры различных способов подхода к табулированной кривой рассмотрены ниже.

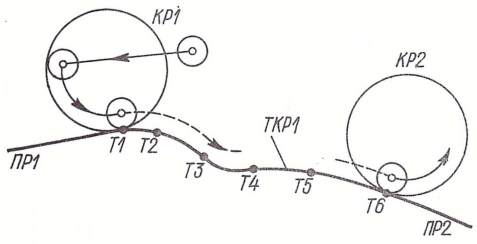


Рис. 2.110

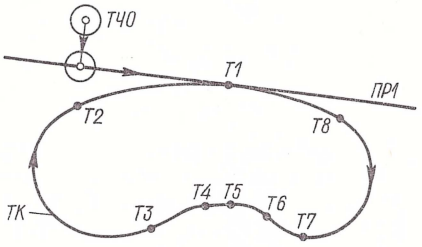


Рис. 2.111

Подход к табулированной кривой и отход от нее *по секущей, не проходящей через граничные точки* (рис. 2.108), является наиболее простым и надежным способом движения по табулированной кривой. Он позволяет исключить начальный и конечный участки сплайна, на которых наиболее вероятно возникновение биений:

```
КОНТУР ТКР1=СПЛАЙН, Т1, Т2, Т3, Т4, Т5, Т6, Т7, Т8
ВПЕРЕД ПР1 ДО ТКР1
ВЛЕВО ТКР1 НА ПР2
ВЛЕВО ПР2 . . .
```

Подход к табулированной кривой *по касательной* осуществляется с помощью прямых *ПР1* и *ПР2*, входящих в геометрическое определение сплайна и задающих касательные к нему в начальной и конечной точках (рис. 2.109):

```
КОНТУР ТКР15=СПЛАЙН, ПР1, Т1, Т2, Т3, Т4, Т5, ПР2
ИДИ НА ПР3
ВЛЕВО ПР3 ДО ПР1
ВЛЕВО ПР1 КАС ТКР15
ВПЕРЕД ТКР15 КАС ПР2
ВПЕРЕД ПР2 НА ПР4
```

Выход в начальную точку табулированной кривой *по дуге окружности* (рис. 2.110) производится в следующей последовательности:

```
КОНТУР ТКР1=СПЛАЙН, ПР1, Т1, Т2, Т3, Т4, Т5, Т6, ПР2
КР1=КАС, ПР1, УБ, Т1, РАДИУС, 20
КР2=КАС, ПР2, УБ, Т6, РАДИУС, 20
. . . . . .
ВЛЕВО КР1 КАС ТКР1
ВПЕРЕД ТКР1 КАС КР2
ВПЕРЕД КР2.
```

При построении *замкнутой табулированной кривой* необходимо в качестве начальной и конечной указывать одну и ту же точку, а также общую касательную, если не допускается излома в этой точке (рис. 2.111):

```
КОНТУР ТК=СПЛАЙН, ПР1, Т1, Т2, Т3, Т4, Т5, Т6, Т7, Т8, Т1, ПР1
ИЗ ТЧ0
ИДИ НА ПР1
ВЛЕВО ПР1 КАС ТК
ВПЕРЕД ТК
```

При *многократных пересечениях с табулированной кривой* (рис. 2.112) для выхода из позиции *1* в позицию *4* (через позиции *2—3*) используются операторы:

ИЗ ТЧ0
ИДИ ДО ПР1
ВЛЕВО ПР1 ДО ТКР
ВЛЕВО ТКР НА 4 ПЕРЕСЕЧ ПР1

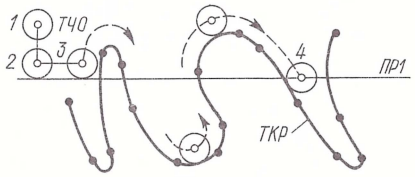


Рис. 2.112

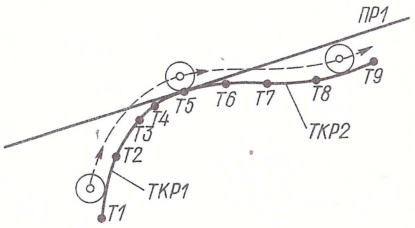


Рис. 2.113

Для *гладкого сопряжения двух табулированных кривых* необходимо задать вырожденное движение по их общей касательной (рис. 2.113):

КОНТУР ТКР1=СПЛАЙН, Т1, Т2, Т3, Т4, Т5, ПР1
КОНТУР ТКР2=СПЛАЙН, ПР1, Т5, Т6, Т7, Т8, Т9
ВПЕРЕД ТКР1 КАС ПР1
ВПЕРЕД ПР1 КАС ТКР2
ВПЕРЕД ТКР2

2.12. Специальные операторы

Управляющие операторы. Эти операторы позволяют изменять порядок выполнения операторов программы, т. е. пропускать (при определенных условиях) или многократно выполнять некоторые ее участки. Такие возможности позволяют составлять на входном языке программы, имеющие сложную логическую структуру. К управляющим операторам относятся операторы НАМЕТКУ, ЕСЛИ и ПОВТОР.

Оператор НАМЕТКУ вызывает безусловную передачу управления на помеченный оператор и имеет следующий формат:

НАМЕТКУ *имя*

где *имя* — имя использованной в программе метки.

После этого оператора выполняется не следующий по порядку оператор, а тот, который помечен указанной меткой. Например:

ИЗ ТЧ8
НАМЕТКУ К2
ИДИ НА ПРХ
К2: В ТОЧКУ ТЧА
ПРИРАЩ 20

В приведенном фрагменте программы после оператора ИЗ будет выполнен не оператор ИДИ, а помеченный оператор ВТОЧКУ.

Оператор ЕСЛИ имеет следующий формат:

ЕСЛИ (*логическое выражение*) *оператор*$_1$ [; *оператор*$_2$; . . . ; *оператор*$_n$]

где *оператор*$_1$, *оператор*$_2$, . . . и т. д.— операторы языка Техтран.

В том случае, если логическое выражение, заключенное в скобки, имеет значение ИСТИНА, то последовательно выполняются *оператор*$_1$, *оператор*$_2$ и т. д. Если значение логического выражения — ЛОЖЬ, то выполняется оператор$_2$, а если он отсутствует, то выполняется оператор, находящийся в следующей строке программы. *Оператор*$_1$, *оператор*$_2$, . . . и т. д. могут быть только простыми операторами входного языка. Применение составных операторов типа ПОВТОР, КОНТУР, МАКРО недопустимо. Например:

ЕСЛИ (А$>$ 0.5 ИЛИ А$<$0) НАМЕТКУ М24

Переход на метку М24 выполняется, если значение переменной А больше 0.5 или меньше 0.

Если значения переменной К меньше 10, то будут выполнены операторы присваивания для ТЧ1 и ТЧ2:

```
ЕСЛИ (К МНШ 10) ТЧ1=0, 0, 5; ТЧ2=ПР3, ПР4
ТЧ6=ЦЕНТР, КР2
```

При других значениях переменной К логическое выражение принимает значение ЛОЖЬ, геометрическое определение точки ТЧ1 опускается и выполняются операторы ТЧ2= ... и ТЧ6= ...

Оператор ЕСЛИ—ТО—ИНАЧЕ—КОНЕСЛИ имеет следующий формат:

```
ЕСЛИ (логическое выражение) ТО
    [список операторов1]
[ИНАЧЕ
    [список операторов2]]
КОНЕСЛИ
```

Если логическое выражение имеет значение ИСТИНА, то последовательно выполняются операторы списка, расположенного после слова ТО. Если же значение логического выражения — ЛОЖЬ, то выполняются операторы списка, стоящего за словом ИНАЧЕ. Любой из этих списков может быть пустым — в этом случае будет выполняться оператор, следующий за ограничителем КОНЕСЛИ. При отсутствии слова КОНЕСЛИ выдается предупреждение и ошибка корректируется. Например:

```
1)  ЕСЛИ (А> =0) ТО
        А=А
    ИНАЧЕ
        А=—А
    КОНЕСЛИ
```

В результате будет вычислено абсолютное значение А.

```
2)  ЕСЛИ (Х> =0) ТО
        У=КВКОР (Х)
    ИНАЧЕ
    КОНЕСЛИ
```

В этом примере отсутствует второй список. Оператор аналогичен оператору ЕСЛИ (Х БРВ 0) У=КВКОР (Х).

```
3) ВПЕРЕД ПРХ НА ПРУ
ПР1=ПАРЛЕЛ, ПРХ, А
ЕСЛИ (А> 0) ТО
    ВЛЕВО ПРУ НА ПР1
ИНАЧЕ
    ВПРАВО ПРУ НА ПР1
КОНЕСЛИ
```

В этом примере выбирается оператор движения ВПРАВО или ВЛЕВО в зависимости от положения прямой ПР1 относительно оси *x*.

Оператор ПОВТОР предназначен для многократного повторения участка программы, называемого телом цикла. Существуют две модификации оператора ПОВТОР: 1) цикл со счетчиком и 2) цикл ПОКА.

Формат 1-го варианта оператора:

```
ПОВТОР перем=нач, кон [, шаг]
оператор1
оператор2
:
операторn
КОНЦИКЛ [перем]
```

где *перем* — идентификатор вещественной переменной, являющейся управляющей переменной цикла; *нач* — число, переменная или арифметическое выражение, задающие начальное значение управляющей переменной цикла; *кон* — число, переменная или арифметическое выражение, задающие конечное значение управляющей переменной цикла; *шаг* — число, переменная, или арифметическое выражение, задающие приращение управляющей переменной; по умолчанию приращение равно 1;

оператор$_1$, *оператор*$_2$, ..., *оператор*$_n$ — операторы входного языка, составляющие тело цикла.

При первом исполнении цикла *перем* = *нач*, при втором — *перем* = *нач* + *шаг* и т. д. до тех пор, пока удовлетворяется условие: *перем***шаг* ⩽ *кон***шаг*. После последнего исполнения цикла управление передается оператору, следующему за оператором КОНЦИКЛ, а управляющая переменная цикла *перем* принимает первое значение, не удовлетворяющее этому условию. Значение управляющей переменной цикла *перем* может быть использовано при вычислениях так же, как значение любой другой вещественной переменной, например:

```
ПОВТОР И=1, 10
А (И)=0
КОНЦИКЛ И
```

В результате выполнения этого цикла первым десяти элементам массива А присваиваются нулевые значения, например:

```
Т=6,5
ПОВТОР И=1,7
Т=Т-И
ЕСЛИ (Т МНШ 0) НАМЕТКУ М1
Т=Т+2.0
КОНЦИКЛ И
М1:Р=Т Т
```

В процессе выполнения цикла переменная *T* последовательно принимает следующие значения: 6,5; 5,5; 7,5; 5,5; 7,5; 4,5; 6,5; 2,5, 4,5; —0,5. Затем оператор ЕСЛИ передает управление на оператор с меткой М1, при этом И=5.

Если начальное значение переменной *T* принять равным 26,5, то последующие ее значения, вычисленные в цикле, будут равны: 27,5; 27,5; 26,5; 24,5; 21,5; 17,5; 12,5. Затем произойдет нормальный выход из цикла; при этом И=8, так как это первое значение переменной И, при котором перестает выполняться условие И*1 ⩽ 7*1.

Формат 2-го варианта оператора:

```
ПОВТОР ПОКА (логическое выражение)
оператор1
оператор2
⋮
операторn
КОНЦИКЛ
```

Такая конструкция позволяет повторять выполнение операторов, составляющих тело цикла, до тех пор, пока логическое выражение, находящееся в заголовке цикла, сохраняет значение ИСТИНА. Для нормального завершения такого цикла необходимо, чтобы в теле цикла имелись операторы присваивания, изменяющие значения переменных, которые входят в логическое выражение. Если перед очередным повторением цикла обнаруживается, что логическое выражение имеет значение ЛОЖЬ, то управление передается оператору, следующему за оператором КОНЦИКЛ. Если значение выражения оказывается ложным с самого начала, то операторы, составляющие цикл, не выполняются.

В качестве примера приведем программу вычисления значения корня кубического с использованием метода Ньютона. Полагаем $x_0=1$ и затем вычисляем,

$$x_1=\frac{2x_0+A/x_0^2}{3},$$

где A — число, из которого необходимо извлечь корень кубический.

Далее полагаем $x_0=x_1$ и вычисляем новое значение x_2. Процесс вычислений продолжается до тех пор, пока разница между двумя последовательными значениями x не окажется в пределах допустимой ошибки. Например, процесс вычисления корня кубического из числа 4 можно представить следующим образом: $A=4$; $x_0=1$; $x_1=(2+4/1)/3=2$; $x_2=(4+4/4)\ 3=1.667$; $x_3=(3.333+4/2.778)/3=1.591$.

Если продолжить процесс вычислений, то значения x будут приближаться к 1,5874 — корню кубическому из 4. Программа на Техтране будет иметь вид:

```
ХСТАР = 0
ХНОВ = 1
ПОВТОР ПОКА ( АБС( ХНОВ — ХСТАР) БЛШ 0.0001)
ХСТАР = ХНОВ
ХНОВ = (2*ХСТАР + 4/ХСТАР**2) /3
КОНЦИКЛ
```

Разрешается использовать вложенные циклы. В этом случае тело внутреннего цикла полностью должно находиться в теле внешнего цикла, т. е. перекрытие циклов не допускается.

Глубина вложений должна быть не больше 10. Рассмотрим случай вложения двух циклов:

```
ВЕЩ А(50,6)
ПОВТОР Н = 1,50
ПОВТОР И = 2,6
ХК = И
А(Н,И) = (1.0 + 0,01*ХК)**И
КОНЦИКЛ И
КОНЦИКЛ Н
```

Эта программа вычисляет значения $A(H,И) = (1+И)^{H}$ для И = 0.01; 0.02; 0.03; ... ...; 0,06 и И = 1, 2, ..., 50.

Не допускается применять операторы, лежащие вне тела цикла и передающие управление внутрь цикла. Варианты допустимых передач управления показаны на рис. 2.114.

Использование циклов должно быть согласовано с использованием макросов. Это означает, что цикл, начинающийся (кончающийся) внутри макроса, должен там же и кончаться (начинаться).

Управление печатью. Каждая фаза процессора системы Техтран осуществляет вывод информации на печать. Для управления объемом выводимой информации в языке имеются специальные операторы: ПЕЧТЕКСТ, ПЕЧТР, СЛЕД и ПЧТ1.

Оператор ПЕЧТЕКСТ предназначен для управления распечаткой исходного текста программы на Техтране (результат работы первой фазы процессора). Оператор имеет следующий формат:

$$\text{ПЕЧТЕКСТ} \left\{ \begin{matrix} \text{ВКЛ} \\ \text{ВСЕ} \\ \text{ВЫКЛ} \end{matrix} \right\}$$

где ВКЛ — распечатка всей вводимой информации кроме расшифровок операторов ВЫЗОВ; ВСЕ — полная распечатка, включая расшифровки операторов ВЫЗОВ; ВЫКЛ — подавление печати вводимой информации.

Отсутствие оператора ПЕЧТЕКСТ эквивалентно применению оператора ПЕЧТЕКСТ ВКЛ. Операторы ДЕТАЛЬ и КОНЕЦ, а также строки, содержащие ошибки, выводятся на печать всегда.

Оператор ПЕЧТР устанавливает режим распечатки траектории движения инструмента и имеет формат:

$$\text{ПЕЧТР} \left\{ \begin{matrix} \text{ВКЛ} \\ \text{ВСЕ} \\ \text{ВЫКЛ} \end{matrix} \right\}$$

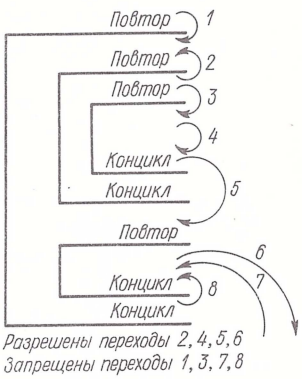


Рис. 2.114

При отсутствии в программе оператора ПЕЧТР или при использовании оператора ПЕЧТР ВКЛ на печать выводятся следующие параметры выполняемого движения: данные об инструменте, метод интерполяции и точность линейной аппроксимации, параметры геометрических объектов, образующих контур, координаты начальной точки, координаты точек пересечения эквидистант, координаты опорных точек.

При наличии в программе таблично заданных кривых и контуров формируются сообщения «НАЧАЛО КОНТУРА», «КОНЕЦ КОНТУРА», а также параметры геометрических элементов и координаты точек пересечения этих элементов.

При использовании оператора ПЕЧТР ВСЕ дополнительно выводятся на печать следующие параметры: координаты точек разбиения окружностей на квадранты, промежуточные точки аппроксимации окружности отрезками прямых, сообщение «ДВИЖЕНИЕ ПО КОНТУРУ» и траектория движения по контуру или по таблично заданной кривой.

Использование оператора ПЕЧТР ВЫКЛ приводит к подавлению вывода информации на печать до конца программы или до следующего оператора ПЕЧТР. Исключение составляют строки, содержащие ошибки движения, которые выводятся всегда.

Данные обо всех геометрических переменных выводятся в соответствии с их внутренним представлением (см. п. 2.9).

Оператор СЛЕД выполняет трассировку программы, т. е. печатает результаты выполнения операторов и имеет следующий формат:

$$\text{СЛЕД} \left\{ \begin{matrix} \text{ВКЛ} \\ \text{ВСЕ} \\ \text{ВЫКЛ} \end{matrix} \right\} \left\{ \begin{matrix} \text{, 1} \\ \text{[, 0]} \end{matrix} \right\}$$

где ВКЛ — включение распечатки результатов выполнения операторов, находящихся вне макроса; ВСЕ — включение распечатки результатов выполнения всех операторов; ВЫКЛ — выключение распечатки результатов выполнения операторов.

Таблица 2.5

Содержание строки	Данные, выводимые на печать
ЕСЛИ (логическое)	ЛОЖЬ или ИСТИНА
НАМЕТКУ	ПЕРЕХОД НА МЕТКУ
ПОВТОР	Минимальное и максимальное значения индекса
КОНЦИКЛ	ПЕРЕХОД НА МЕТКУ
ВЫЗОВ	ВХОД В МАКРО
КОНЕЦ МАКРОСА	ВЫХОД ИЗ МАКРО
Арифметическое выражение	Результат
Геометрическое определение	Имя параметра геометрического объекта
КОНТУР	Параметры геометрических объектов, образующих контур; координаты опорных точек табулированной кривой
Примечание. Перед результатом на печать выводится номер соответствующей строки.	

В режиме «1» печатаются результаты всех операторов, приведенных в табл. 2.5. В режиме «0» печатаются только результаты управляющих операторов. Вся информация выводится в процессе работы второй фазы процессора в формате, приведенном в табл. 2.5.

Оператор ПЧТ1 используется для печати значения некоторой переменной или параметров геометрического объекта (точки, окружности, вектора или прямой) и имеет следующий формат:

ПЧТ1 $имя_1$, $имя_2$, $имя_3$, ...

где $имя_1$, $имя_2$, ... — имена геометрических объектов или переменных.

Аналогично оператору СЛЕД информация выводится в процессе работы второй фазы процессора в следующем формате:

номер строки $имя_1$ = внутреннее представление

номер строки $имя_2$ = внутреннее представление

Оператор СИНОНИМ служит для замены стандартных служебных слов синонимами и имеет следующий формат:

СИНОНИМ (*служебное слово*$_1$, *синоним*$_1$), %
[(*служебное слово*$_2$, *синоним*$_2$), %
:
:
(служебное слово$_n$, синоним$_n$)]

В качестве *синонимов* могут употребляться только имена. Оператор должен быть задан до первого использования синонима. Например, задав оператор СИНОНИМ (ВПЕРЕД, В), (ПЕРЕСЕЧ, П), можно вместо оператора ВПЕРЕД ПР1 ДО 2 ПЕРЕСЕЧ КР1 употреблять в программе оператор В ПР1 ДО 2 П КР1.

Оператор вставки текста «*» позволяет вставить в программу текст, хранящийся во внешней памяти, и имеет следующий формат:

* *имя*
* *литерал*

где *имя* — имя модуля из библиотеки программ; *литерал* — спецификация файла. Например:

* PROG1
*'SY:PROG2'

Конкретный формат оператора зависит от типа используемой ЭВМ.

2.13. Макросы

В языке Техтран можно использовать в программе отдельно подготовленные части исходного текста — макросы. В макросе разрешается применять любые операторы языка, но вместо конкретных значений переменных могут использоваться параметры.

Подключение макроса к программе осуществляется при помощи специального оператора ВЫЗОВ. При этом в тексте программы на месте оператора ВЫЗОВ появляется текст вызываемого макроса (тело), в котором формальные параметры, заданные в теле макроса, заменяются фактическими значениями, указанными в операторе ВЫЗОВ. Допускается определение макросов внутри программы. В этом случае вызов макроса должен производиться из этой же программы.

Структура макросов. Макрос состоит из заголовка, тела и ограничителя. Заголовок макроса представляет собой специальный оператор МАКРО. Тело макроса — произвольная последовательность операторов языка Техтран (кроме операторов ДЕТАЛЬ, КОНЕЦ и ПЕЧТЕКСТ), в которых могут быть использованы параметры макроса. Тело макроса может содержать другие макросы и обращения к ним. Ограничителем макроса является служебное слово КОНМАК.

Параметры макроса. Параметры могут быть операторными и текстовыми. Параметры записываются в произвольном порядке и отделяются друг от друга запятыми.

Операторный параметр представляет собой оператор языка Техтран, заключенный в скобки. При вызове макроса операторные параметры выполняются непосредственно перед телом макроса. В качестве операторных параметров нельзя использовать операторы ДЕТАЛЬ, КОНЕЦ, МАКРО, КОНМАК, ВЫЗОВ, КОНТУР, КОНКОН, ПОВТОР, КОНЦИКЛ, ЕСЛИ. Операторы присваивания, не содержащие разделителей в правой части, в скобки можно не заключать.

Текстовой параметр в отличие от операторного имеет свое имя и значение. Значением текстового параметра является произвольная последовательность символов, не содержащая символа «#». Имя текстового параметра — произвольное имя, допустимое в языке Техтран.

В теле макроса имени текстового параметра должен предшествовать символ макроподстановки «&». При вызове макроса вместо символа «&» и имени параметра в текст будет подставлено его значение. Не разрешается параметризовать часть имени, числа или литерала.

Например, вместо оператора КР& КР=0,0,40 после текстовой подстановки значения параметра '10' получим: КР 10=0,0,40, что недопустимо.

Правильной является запись: &КР=0,0,40. При значении параметра 'КР10' оператор будет преобразован к виду: КР10=0,0,40.

Значения текстовых параметров макроса задаются в операторах ВЫЗОВ или МАКРО.

Заголовок макроса. Заголовок представляет собой специальный оператор МАКРО и имеет следующий формат:

МАКРО *имя* [(*список значений*)]

где *имя* — имя макроса (любое имя, допущенное в языке Техтран); *список значений* — список значений текстовых параметров, установленных по умолчанию.

Если значения по умолчанию не задаются, скобки должны быть опущены. Значение по умолчанию используют в том случае, если в операторе ВЫЗОВ значение данного текстового параметра не задано. Элементы списка значений отделяются друг от друга запятыми.

Значение по умолчанию задается одним из следующих способов:

1) *'имя'=литерал* (например, 'КР'='КР10' — параметр с именем КР принимает по умолчанию значение, равное КР10);

2) *'имя$_1$'=& имя$_2$* (например, 'КР1'=&КР2 — параметр с именем КР1 принимает то же значение, что и параметр КР2; параметр КР2 должен быть определен ранее);

3) *'имя'=объединение литералов и текстовых параметров* (например, если значение параметра КР по умолчанию определено следующим образом: 'КР'= ='КР'&М6&АБ'10', то при значениях параметров М6 и АБ'04' и 'В' соответственно, параметр с именем КР примет значение КР04В10).

Примеры заголовков макросов:

МАКРО МТК1

Значений по умолчанию нет.

МАКРО М2 ('П1'='1',' П2'='ПР'&П1,' П3'='КР'&П1&П1)

По умолчанию: П1=1; П2=ПР1; П3=КР11

МАКРО М3 ('П1'='1','П2'='ПР'&П4&П4',П3'='КР'&П4)

По умолчанию: П1=1; П2=ПР *значение* П4 *значение* П4; П3=КР *значение* П4.

Значение параметра П4 обязательно должно быть определено при вызове макроса. В противном случае возникнет ошибка. Если для параметра П4 нужно задать значение по умолчанию, то это следует сделать до присваивания значений П2 и П3: МАКРО П3 ('П1'='1', 'П4'='2', 'П2'='ПР'&П4&П4, % 'П3'='КР'&П4)

Вызов макроса. Для выполнения макроподстановки, т. е. вставки в текст исходной программы на Техтране тела макроса и для замены формальных параметров фактическими предназначен оператор ВЫЗОВ. Он имеет следующий формат:

ВЫЗОВ $\left\{\begin{matrix}\textit{имя}\\ \textit{литерал}\end{matrix}\right\}$ [(*список параметров*)]

где *имя* — имя макроса, использованное в операторе МАКРО; *литерал* — спецификация файла, в котором находится текст макроса; *список параметров* — последовательность операторных параметров и значений текстовых параметров, разделенных запятыми. Если параметры в вызове отсутствуют, скобки должны быть опущены.

Имя макроса используется в том случае, когда макрос находится в макробиблиотеке или определяется в самой программе. Если макрос содержится в отдельном файле, следует использовать литерал — спецификацию этого файла. Формат спецификации файла зависит от типа ЭВМ и операционной системы, например:

ВЫЗОВ М1

ВЫЗОВ 'SY:MACRO 1'

С помощью оператора «:» (двоеточие) можно задавать умолчание для имени макроса. Формат оператора следующий:

: *литерал$_1$*, *литерал$_2$*

где *литерал$_1$* — строка символов, присоединяемая к имени макроса слева; *литерал$_2$* — строка символов, присоединяемая к имени макроса справа.

Например, оператор: 'DK1: [77, 300]', '. TEH' позволяет, задавая в программе вызов в формате ВЫЗОВ МАКРО1, передать операционной системе имя файла в виде 'DK1: [77, 300] MAKRO 1. TEH'.

Рассмотрим следующий оператор:

ВЫЗОВ М1 (А=1, (ВЕЩ Б=2), В=30.5, (ТЧ1=0,0,0))

В нем заданы четыре операторных параметра. Без скобок можно записывать только параметры, присваивающие вещественные значения.

Если в операторе ВЫЗОВ определено значение текстового параметра, которому присвоено значение по умолчанию в заголовке макроса, то значение по умолчанию игнорируется, например:

```
ДЕТАЛЬ 'МАКРО'
МАКРО М1 ('П1'='1', 'П2'=& П1'0')
    А=&П1+& П2
    ТЧ1=А, В
    КР1=ТЧ1, В+& П1
    ПЧТ1 А, В, ТЧ1, КР1
КОНМАК
ВЫЗОВ М1 ('П1'='2', 'П2'='25', В=0)
ВЫЗОВ М1 ('П1'='2', В=3)
ВЫЗОВ М1 ('П2'='35', В=2.5)
ВЫЗОВ М1 (В=7.4)
КОНЕЦ
```

Таблица 2.6

Номер вызова макроса	П1	П2	В	А	ТЧ1			КР1		
1	2	25	0	27	27	0	0	27	0	2
2	2	20	3	22	22	3	0	22	3	5
3	1	35	2.5	36	36	2.5	0	36	2.5	3.5
4	1	10	7.4	11	11	7.4	0	11	7.4	8.4

Значения параметров и переменных, выводимые на печать, приведены в табл. 2.6.

Вложенные вызовы. В теле макроса может встретиться оператор ВЫЗОВ с именем другого макроса. При этом происходит вставка тела нового макроса в тело старого. В теле внутреннего макроса нельзя использовать текстовые параметры внешнего макроса, но их можно применять в операторе ВЫЗОВ для задания значений внутренних текстовых параметров, например:

```
МАКРО М1 ('П1'='', П2='', П3='')
    ВЫЗОВ М2 ('П1'='&П1')
    ВЫЗОВ М2 ('П1'='&П2+&П3')
КОНМАК
МАКРО М2
    А=&П1—1
КОНМАК
ВЫЗОВ М1 ('П1'='10')
ВЫЗОВ М1 ('П2'='1', 'П3'='2')
```

Таблица 2.7

Номер вызова макроса М2	П1	А
1	10	А=10—1
2	+	А=+—1
3	—	А=—1
4	1+2	А=1+2—1

Примечание. В результате 2-го вызова макроса будет выдана ошибка компиляции.

Значения параметра и переменной, выводимые на печать (для данного примера), приведены в табл. 2.7.

Локализация переменных в макросе. Для обеспечения полной независимости макросов от вызывающей программы предусмотрена возможность локализации переменных внутри макросов.

Переменная является локализованной в макросе, если ее значение доступно только внутри этого макроса. В этом случае вне макроса может существовать другая переменная с тем же именем.

Для локализации переменной внутри макроса ее необходимо там явно описать. В противном случае будет использоваться переменная, определенная вне макроса. Если переменная используется впервые, она считается локализованной на текущем уровне вложения макросов.

Приведем пример использования локальных переменных:

```
A=1
МАКРО М
    ВЕЩ А
    А=2
КОНМАК
ВЫЗОВ М
ПЧТ1 А
```

Поскольку в макросе М имеется явное задание типа переменной А, то она считается локальной в макросе М. Вне макроса ее значение, равное 2, недоступно. Вне тела макроса определена другая переменная А, которой было присвоено значение, равное 1, и именно это значение будет выведено на печать:

```
A=1
МАКРО М
    А=2
КОНМАК
ВЫЗОВ М
ПЧТ1 А
```

В этом случае печатается значение А=2, так как в теле макроса используется переменная, определенная вне его. Расположение тела макроса относительно оператора А=1 здесь не существенно:

```
МАКРО М
    А=2
КОНМАК
ВЫЗОВ М
ПЧТ1 А
```

В данном примере допущена ошибка, так как значение переменной А не определено.

Операторные параметры рассматриваются как операторы, находящиеся внутри макроса.

Переменные, определяемые в операторных параметрах, должны удовлетворять тем же соглашениям, что и переменные тела макроса, например:

```
A=1
ВЫЗОВ М (ВЕЩ А=2)
ПЧТ1 А
```

На печать будет выведено А=1, так как в вызове определена локальная в макросе М переменная.

```
A=1
ВЫЗОВ М (А=2)
ПЧТ1 А
```

На печать будет выведено значение А=2.

Использование меток. Все используемые в макросе метки локализованы внутри этого макроса. Это означает, что передачи управления в макрос и из него запрещены. Разрешены только переходы внутри тела макроса.

2.14. Примеры программ на Техтране

Примеры, приведенные ниже, иллюстрируют возможности процессора САП Техтран. Поэтому в них, как правило, отсутствуют операторы постпроцессора и некоторые вспомогательные перемещения, необходимые с точки зрения технолога.

Пример 1. Ниже приводится текст программы обработки простейшего контура, изображенного на рис. 2.115.

```
ДЕТАЛЬ 'ПРИМЕР 1'
#
#    ОПРЕДЕЛЕНИЕ ЭЛЕМЕНТОВ КОНТУРА
#
ТЧ0=-30,0
ТЧ1=0,0
ПР1=ТЧ1,45
ПР2=ТЧ1,-45
ТЧ2=ПР1,ТЧ1,50,УМ
ТЧ3=ПР2,ТЧ1,50,УМ
ПР3=ТЧ0,ПЕРП,ПР1
КР1=ТЧ1,ТЧ2,ТЧ3
КР2=УМ,ПР1,УМ,ПР2,РАДИУС,10
КР3=УМ,ПР1,ХМ,ВНУТРИ,КР1,10
КР4=УМ,ПР2,ХБ,ВНУТРИ,КР1,10
#
ИНСТР 6                          # ДИАМЕТР ИНСТРУМЕНТА
МЕТОД ЛИНКРУГ                    # МЕТОД ИНТЕРПОЛЯЦИИ
ИНСТПР                           # ИНСТРУМЕНТ СПРАВА
#
#    ОПИСАНИЕ ТРАЕКТОРИИ ДВИЖЕНИЯ ИНСТРУМЕНТА
#
ИЗ ТЧ0                           # НАЧАЛО ДВИЖЕНИЯ
ИДИ ДО ПР1                       # ПОДХОД К КОНТУРУ
ВПРАВО ПР1                       # ОБХОД КОНТУРА
ВПЕРЕД КР3:КР1:КР4:ПР2: %        #  СНАРУЖИ ПРОТИВ
       КР2:ПР1 НА ПР3            #  ЧАСОВОЙ СТРЕЛКИ
ВТОЧКУ ТЧ0                       # ВОЗВРАТ В ИСХ. ТОЧКУ
КОНЕЦ                            # КОНЕЦ ПРОГРАММЫ
```

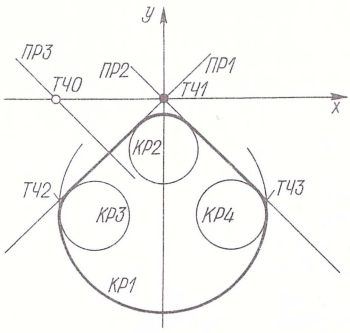


Рис. 2.115

Пример 2. Ниже приведен пример использования матриц симметрии. Контур, изображенный на рис. 2.116, имеет две оси симметрии *x* и *y*. В качестве базового выбран элемент, лежащий в 4-м квадранте. Обход элементов, находящихся в 1 и 3-м квадрантах, задается симметричным отображением траектории обработки базового элемента относительно осей *x* и *y*

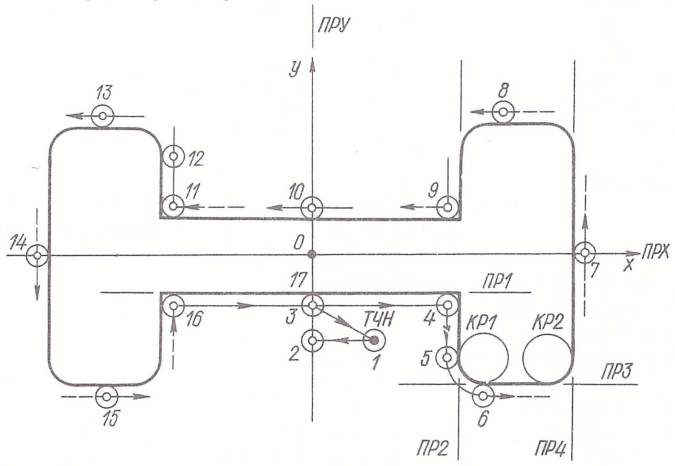


Рис. 2.116

соответственно. Преобразование траектории для обработки части контура, лежащей во 2-м квадранте, осуществляется по матрице М3, являющейся произведением матриц симметрии относительно осей *x* и *y*. Программа имеет следующий вид:

```
ДЕТАЛЬ 'ПРИМЕР2'
ПРХ=ХПАР,0
ПРУ=УПАР,0
ПР1=ХПАР,-10
ПР2=УПАР,40
ПР3=ХПАР,-35
ПР4=УПАР,70
```

```
КР1=ХБ,ПР2,УБ,ПР3,5
КР2=ХМ,ПР4,УБ,ПР3,5
ТЧН=10,-20
МАТР М1,М2,М3
М1=ПРХ                  # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "Х"
М2=ПРУ                  # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "У"
М3=М1,М2                # ПРОИЗВЕДЕНИЕ МАТРИЦ СИММЕТРИИ
ИНСТПР
ИНСТР 3
ИЗ ТЧН                                  # ПОЗ. 1
ИДИ НА ПРУ                              # ПОЗ. 1-2
ВПРАВО ПРУ ДО ПР1                       # ПОЗ. 2-3
ВПРАВО ПР1:ПР2                          # ПОЗ. 3-4-5
ВПЕРЕД КР1:ПР3:КР2:ПР4 НА ПРХ           # ПОЗ. 5-6-7
ТРАНС М1
ВПЕРЕД ПР4:КР2:ПР3:КР1:ПР2              # ПОЗ. 7-8-9
ВЛЕВО ПР1 НА ПРУ                        # ПОЗ. 9-10
ТРАНС М3
ВПЕРЕД ПР1                              # ПОЗ. 10-11
ВПРАВО ПР2                              # ПОЗ. 11-12
ВПЕРЕД КР1:ПР3:КР2:ПР4 НА ПРХ           # ПОЗ. 12-13-14
ТРАНС М2
ВПЕРЕД ПР4:КР2:ПР3:КР1:ПР2              # ПОЗ. 14-15-16
ВЛЕВО ПР1 НА ПРУ                        # ПОЗ. 16-17
ТРАНС ОТМЕН
ВТОЧКУ ТЧН
КОНЕЦ
```

Программу можно написать и с использованием объектов типа КОНТУР:

```
ДЕТАЛЬ 'ПРИМЕР2А'
ПРХ=ХПАР,0
ПРУ=УПАР,0
ПР1=ХПАР,-10
ПР2=УПАР,40
ПР3=ХПАР,-35
ПР4=УПАР,70
КР1=ХБ,ПР2,УБ,ПР3,5
КР2=ХМ,ПР4,УБ,ПР3,5
ТЧ1=ПР1,ПРУ
ТЧН=10,-20
МАТР М1,М2,М3
М1=ПРХ                  # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "Х"
М2=ПРУ                  # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "У"
М3=М1,М2                # ПРОИЗВЕДЕНИЕ МАТРИЦ СИММЕТРИИ
# ИСХОДНЫЙ КОНТУР (В 4 КВАДРАНТЕ)
КОНТУР К=ИЗ,ТЧ1,УБ
   ВПРАВО ПР1:ПР2
   ВПЕРЕД КР1:ПР3:КР2:ПР4 НА ПРХ
КОНКОН
# СОСТАВНОЙ КОНТУР
КОНТУР К1=       К,  %     # 4 КВАДРАНТ
          -1,М1,К,   %     # 1 КВАДРАНТ
             М3,К,   %     # 2 КВАДРАНТ
          -1,М2,К          # 3 КВАДРАНТ
ИНСТПР
ИНСТР 3
ИЗ ТЧН
ИДИ НА ПРУ
ВПРАВО ПРУ ДО К1        # ПОДХОД К КОНТУРУ
ВПРАВО К1               # ОБХОД КОНТУРА
ВТОЧКУ ТЧН
КОНЕЦ
```

Пример 3. Этот пример иллюстрирует совместное применение матриц симметрии и переноса, а также использование оператора ПОВТОР (рис. 2.117). Следует обратить внимание на порядок записи матриц-сомножителей. Произведение матриц не коммутативно, поэтому иной

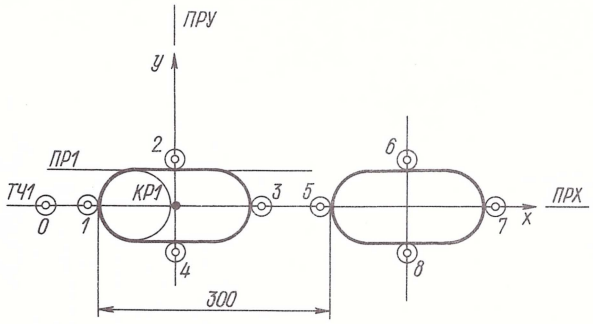


Рис. 2.117

порядок их записи приведет к ошибке. При первом выполнении тела цикла обрабатывается левый элемент без смещения, так как преобразование по матрице М4 не вызывает изменения координат.

При втором выполнении цикла обрабатывается правый элемент.

```
ДЕТАЛЬ 'ПРИМЕР3'
ПРХ=ХПАР,0
ПРУ=УПАР,0
КР1=-50,0,50
ПР1=ХПАР,50
ТЧ1=-150,0
СДВ=300
МАТР М1,М2,М3,М4
М1=ПРХ                          # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "Х"
М2=ПРУ                          # СИММЕТРИЯ ОТНОСИТЕЛЬНО ОСИ "У"
М3=М1,М2                        # ПРОИЗВЕДЕНИЕ МАТРИЦ СИММЕТРИИ
ИНСТР 10
ИЗ ТЧ1
ИНСТЛВ
ПОВТОР И=1,2                          #   УЧАСТОК ТРАЕКТОРИИ:
   М4=ПЕРЕНОС,(И-1)*СДВ,0             #   ПРИ И=1     ПРИ И=2
   ТРАНС М4
   ИДИ ДО КР1                         #       0-1         1-5
   ВЛЕВО КР1
   ВПЕРЕД ПР1 НА ПРУ                  #       1-2         5-6
   ТРАНС МАТР(М4,М2)
   ВПЕРЕД ПР1:КР1 НА ПРХ              #       2-3         6-7
   ТРАНС МАТР(М4,М3)
   ВПЕРЕД КР1:ПР1 НА ПРУ              #       3-4         7-8
   ТРАНС МАТР(М4,М1)
   ВПЕРЕД ПР1:КР1 НА ПРХ              #       4-1         8-5
КОНЦИКЛ
ТРАНС ОТМЕН
КОНЕЦ
```

Пример 4. Приводится текст программы, в которой используются матричные преобразования и вложенные циклы. Запрограммирована обработка 32 элементов, полученных сдвигом (рис. 2.118, *а*) и поворотом базового элемента (рис. 2.118, *б*). Траектория обработки задана с помощью двух вложенных циклов: внешний цикл — поворот на 0, 45, 90°, ... и т. д.; внутренний цикл — перенос на 0, 40, 80 и 120 мм.

Первое выполнение тела цикла обеспечивает обход четырех элементов без их разворота, затем величина сдвига сбрасывается и внутренний цикл выполняется для нового угла поворота, отличного от нуля; элементы обрабатываются в следующем порядке: *1—2—3—4—5—6—7—8—9—10* ... и т. д.

```
ДЕТАЛЬ 'ПРИМЕР4'
МАТР М,МПЕРЕНОС,МПОВОРОТ
ТЧЦ=0,0
ТЧ0=110,0
ПРХ=ХПАР,0
ПР1=УПАР,100
ТЧ1=110,20
ТЧ2=110,-20
КР1=ЦЕНТР,ТЧ1,КАС,ПР1
КР2=ТЧ2,ПР1
ПР2=СЛЕВА,КР1,СЛЕВА,КР2
МПЕРЕНОС=ПЕРЕНОС,0,0
МПОВОРОТ=ПОВОРОТ,0
ИНСТР 5
ИЗ ТЧЦ
ИНСТПР
ПОВТОР НОМПОВ=1,8
   ПОВТОР НОМПЕР=1,4
      М=МПОВОРОТ,МПЕРЕНОС
      ТРАНС М
      ВТОЧКУ ТЧ0
      ПРИРАЩ -50
      ИДИ ДО ПР1
      ВПРАВО ПР1
      ВПЕРЕД КР1:ПР2:КР2:ПР1 НА ПРХ
      ПРИРАЩ 50
      ВТОЧКУ ТЧ0
      МПЕРЕНОС=ПЕРЕНОС,40*НОМПЕР,0
   КОНЦИКЛ НОМПЕР
   МПОВОРОТ=ПОВОРОТ,45*НОМПОВ
   МПЕРЕНОС=ПЕРЕНОС,0,0
КОНЦИКЛ НОМПОВ
ТРАНС ОТМЕН
ВТОЧКУ ТЧЦ
КОНЕЦ
```

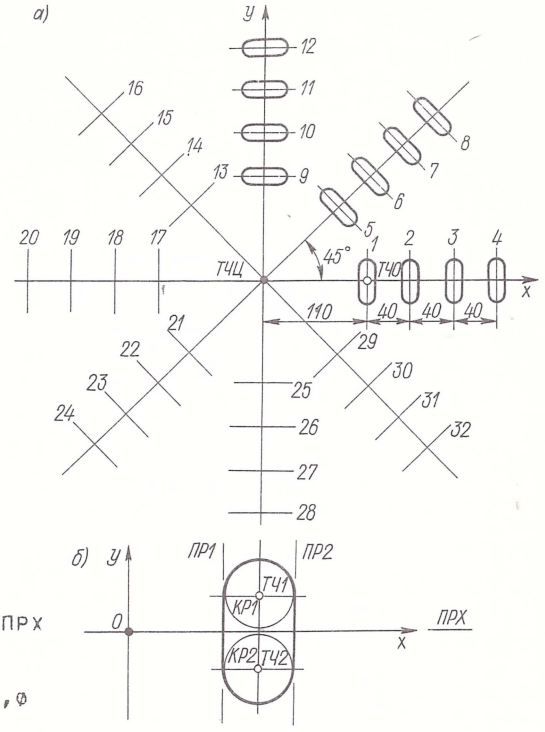


Рис. 2.118

Пример 5. Этот пример иллюстрирует применение табулированных кривых. Контур детали представляет собой сложную кривую, заданную в параметрическом виде (рис. 2.119):

$$y=\frac{1}{2}\sqrt{k_1\cos^2 t+k_2\cos t};$$
$$x=y\operatorname{tg} t$$

где $k_1=12\,544$; $k_2=9956$ для $0\leqslant x\leqslant 250$ и $0\leqslant t\leqslant 89°$.

```
ДЕТАЛЬ 'ПРИМЕР5'
СЛЕД ВКЛ,1
ВЕЩ ПРИРУГ
ТОЧКА Т(100)
ЧИСЛТОЧ=0          # ЧИСЛО  ТОЧЕК
УГОЛП=0            # ТЕКУЩЕЕ ЗНАЧЕНИЕ ПАРАМЕТРА
ПРИРУГ=1           # ПРИРАЩЕНИЕ ПАРАМЕТРА
# В ЦИКЛЕ ВЫЧИСЛЯЮТСЯ КООРДИНАТЫ ТОЧЕК, ЛЕЖАЩИХ В
# ЗАДАННОМ ДИАПАЗОНЕ ИЗМЕНЕНИЯ ПАРАМЕТРА "УГОЛП".
ПОВТОР ПОКА УГОЛП<=88
   ЧИСЛТОЧ=ЧИСЛТОЧ+1
   У=0.5*КВКОР(12544*(КОС(УГОЛП))**2+9956*КОС(УГОЛП))
   Х=У*ТАН(УГОЛП)
   Т(ЧИСЛТОЧ)=Х,У
   ЕСЛИ( УГОЛП>80 ) ПРИРУГ=0.5
   #   ПО ЭТОМУ УСЛОВИЮ ИЗМЕНЯЕТСЯ ПРИРАЩЕНИЕ
   #   УГЛА НА ПОЛОГОМ УЧАСТКЕ КРИВОЙ
   УГОЛП=УГОЛП+ПРИРУГ
```

```
КОНЦИКЛ
ПР1=ХПАР,0
ПР2=УПАР,0
ПР3=УПАР,250
ПР4=Т(1),0
ТЧН=-10,100
КОНТУР ТТ=СПЛАЙН,ПР4
   ПОВТОР И=1,ЧИСЛТОЧ
      Т(И)
   КОНЦИКЛ
КОНКОН
ИЗ ТЧН
ИНСТР 0
ИДИ ДО ПР4
ВЛЕВО ПР4 ДО ТТ
ВПЕРЕД ТТ НА ПР3
ВТОЧКУ ТЧН
КОНЕЦ
```

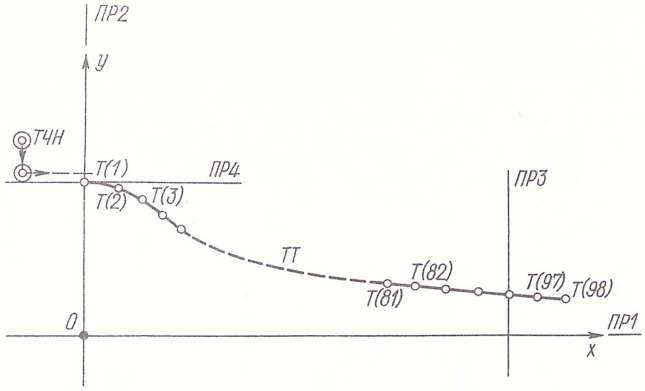


Рис. 2.119

Пример 6. На данном примере показано использование в программе объектов типа КОНТУР (рис. 2.120):

```
ДЕТАЛЬ 'ПРИМЕР6'
ТЧ1=290,120
ПР1=ХПАР,40
ПР2=УПАР,40
ПР3=УПАР,80
ПР4=УПАР,120
ПР5=УПАР,160
ПР6=УПАР,200
ПР7=УПАР,240
ПР8=ТОЧКА(0,110),10
ПР9=ХПАР,200
ПР10=ХПАР,80
КР1=ХБ,ПР2,УМ,ПР9,ХМ,ПР3
КР2=ХБ,ПР3,УБ,ПР10,ХМ,ПР4
КР3=ХБ,ПР4,УМ,ПР9,ХМ,ПР5
КР4=ХБ,ПР5,УБ,ПР10,ХМ,ПР6
КР5=ХБ,ПР6,УМ,ПР9,ХМ,ПР7
КР6=ХМ,ПР7,УБ,ПР1,20
КР7=ХБ,ПР2,УБ,ПР1,20
ТЧ2=150,40
#   КОНТУР "К" - ЗАМКНУТЫЙ КОНТУР  С НАЧАЛОМ В ТОЧКЕ "ТЧ2"
КОНТУР К=ИЗ,ТЧ2,ХМ
   ВПЕРЕД ПР1:КР7:ПР2:КР1:ПР3
   ВПЕРЕД КР2:ПР4:КР3:ПР5:КР4
   ВПЕРЕД ПР6:КР5:ПР7:КР6:ПР1
КОНКОН
ИНСТР 8
ИЗ ТЧ1
ИДИ ДО ПР8                    # ПОЗ. 1-2
ВЛЕВО ПР8 ДО 3 ПЕРЕСЕЧ К     # ПОЗ. 2-3
ВПРАВО К ЗА 6 ПЕРЕСЕЧ ПР8    # ПОЗ. 3-4-5-6
ВТОЧКУ ТЧ1
КОНЕЦ
```

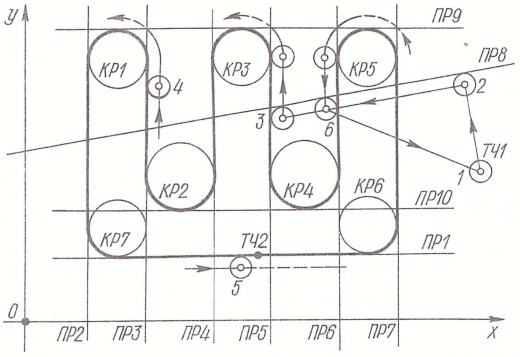


Рис. 2.120

Пример 7. На рис. 2.121, *а* изображена траектория сверления двух глубоких отверстий, каждое из которых обрабатывается по циклу глубокого сверления с выводом инструмента (рис. 2.121, *б*). Программа содержит два вызова макроса ГЛУБСВ, заданного в теле программы и определяющего траекторию глубокого сверления одного отверстия.

Операторный параметр вида ТОЧКА ТЧ=50, 100, 30 находящийся в первом вызове, определяет переменную ТЧ как локальную в макросе ГЛУБСВ. Благодаря этому другая переменная с таким же именем, но определенная в головной программе, сохраняет свое первоначальное

значение после выхода из макроса. Остальные параметры будут локальными переменными макроса, если переменные с такими же именами не будут использоваться в головной программе.

Параметр ПЛБЕЗ задан как глобальная переменная, значение которой доступно в макросе. Поэтому для каждого из вызовов начальное и конечное положение инструмента по координате z равны 100 мм. Параметр НЕДОХОД — текстовой. Его значение по умолчанию равно 2, что

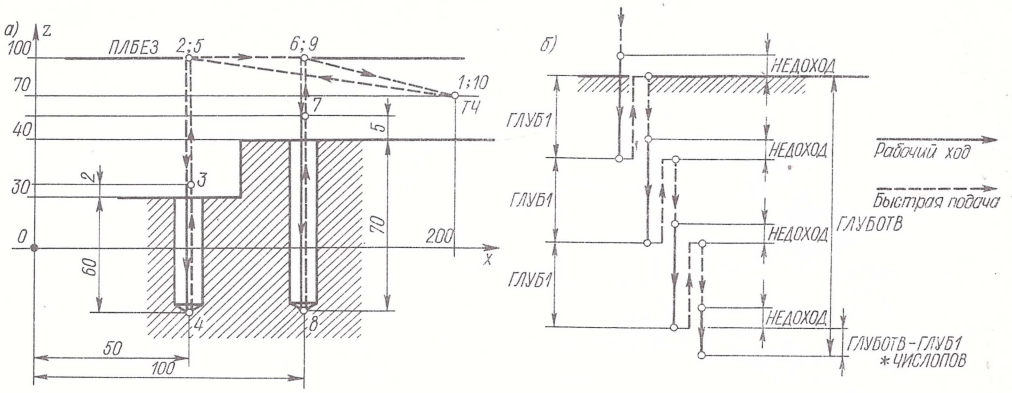


Рис. 2.121

определено в заголовке макроса, поэтому для первого из вызовов недоход до поверхности детали составит 2 мм, а для второго будет равен значению параметра НЕДОХОД, заданному в вызове, т. е. 5 мм. Программа имеет следующий вид.

```
ДЕТАЛЬ 'ПРИМЕР7'
СТАНОК 3Ø9
ПЕЧТЕКСТ ВСЕ
СЛЕД ВСЕ,1
МАКРО ГЛУБСВ('НЕДОХОД'='2')
#
#          Г Л У Б О К О Е   С В Е Р Л Е Н И Е
#
# ПАРАМЕТРЫ МАКРОСА :
#  ТЧ       - ТОЧКА, В КОТОРОЙ НУЖНО ВЫПОЛНИТЬ ЦИКЛ
#  ПЛБЕЗ    - ПЛОСКОСТЬ БЕЗОПАСНОСТИ; РАССТОЯНИЕ ПЛОСКОСТИ
#             БЕЗОПАСНОСТИ ОТ ПЛОСКОСТИ ХОУ
#  НЕДОХОД  - ВЕЛИЧИНА НЕДОХОДА ДО МАТЕРИАЛА;
#             ПО УМОЛЧАНИЮ = 2ММ
#  ГЛУБОТВ  - ГЛУБИНА ОТВЕРСТИЯ
#  ГЛУБ1    - ГЛУБИНА ОДНОГО ПРОХОДА
#  ПОД      - ПОДАЧА В ЦИКЛЕ
#
# ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ :
#  ЧИСЛПОВ  - ЧИСЛО ВРЕЗАНИЙ НА ГЛУБИНУ ГЛУБ1
#  ТЧ1      - ТОЧКА НА ПЛОСКОСТИ БЕЗОПАСНОСТИ, ИЗ КОТОРОЙ
#             НАЧИНАЕТСЯ И В КОТОРОЙ ЗАКАНЧИВАЕТСЯ ЦИКЛ
#
   ВЕЩ ЧИСЛПОВ                      #      ОПИСАНИЕ
   ТОЧКА ТЧ1                        # ЛОКАЛЬНЫХ ПЕРЕМЕННЫХ
   ТЧ1=КООРД(1,ТЧ),КООРД(2,ТЧ),ПЛБЕЗ
   БЫСТРО
   ВТОЧКУ ТЧ1                       # ВЫХОД В ПЛ. БЕЗОПАСНОСТИ
   ПРИРАЩ -(ПЛБЕЗ-(КООРД(3,ТЧ)+&НЕДОХОД))
   ЧИСЛПОВ=ЦЕЛ(ГЛУБОТВ/ГЛУБ1)       # ЧИСЛО ВРЕЗАНИЙ
   ПОВТОР И=1,ЧИСЛПОВ               # НА ГЛУБИНУ ГЛУБ1
      ПОДАЧА ММИН,ПОД
      ПРИРАЩ -(ГЛУБ1+&НЕДОХОД)      # ВРЕЗАНИЕ
```

```
      БЫСТРО
      ПРИРАЩ ГЛУБ1                    # ПОДЪЕМ
      ПРИРАЩ -(ГЛУБ1-&НЕДОХОД)        # ОПУСКАНИЕ
   КОНЦИКЛ
   ПОДАЧА ММИН,ПОД                    # ПОСЛЕДНЕЕ ВРЕЗАНИЕ
   ПРИРАЩ  -(ГЛУБОТВ-ГЛУБ1*ЧИСЛПОВ+&НЕДОХОД)
   БЫСТРО
   ВТОЧКУ ТЧ1                         # ВЫХОД В ПЛ. БЕЗОПАСНОСТИ
КОНМАК
#
ТЧ=200,100,70
ТЧ100=100,100,40
ИЗ ТЧ                                 # НАЧАЛО ДВИЖЕНИЯ
ПЛБЕЗ=100                             # ГЛОБАЛЬНАЯ ПЕРЕМЕННАЯ;
                                      # ЕЕ ЗНАЧЕНИЕ ДОСТУПНО В
                                      # МАКРОСЕ
ВЫЗОВ ГЛУБСВ((ТОЧКА ТЧ=50,100,30),ГЛУБОТВ=60, %
             ГЛУБ1=18,ПОД=50)
ВЫЗОВ ГЛУБСВ((ТОЧКА ТЧ=ТЧ100),'НЕДОХОД'='5', %
             ГЛУБОТВ=60,ГЛУБ1=18,ПОД=70)
ВТОЧКУ ТЧ
КОНЕЦ
```

Глава 3

ОПЕРАТОРЫ ПОСТПРОЦЕССОРА

3.1. Общие положения

Операторы постпроцессора задают режимы обработки детали на станке с ЧПУ и управляют постпроцессором. Их используют для задания подачи, частоты вращения шпинделя, коррекции, охлаждения и т. д. Как отмечалось ранее, операторы постпроцессора не обрабатываются процессором. Синтаксический и семантический контроль их выполняется постпроцессором. Ниже приводится описание основных операторов, реализованных в постпроцессорах САП Техтран. В описании каждого оператора, как правило, даны несколько форматов, используемых в различных постпроцессорах.

При описании синтаксиса операторов постпроцессора применяются те же правила, что и в гл. 2. Общий формат операторов постпроцессора имеет вид:

главное слово { *вспомогательное слово* / *литерал* / *число* }, ...

Главное слово идентифицирует оператор постпроцессора. Операндами операторов постпроцессора могут быть служебные слова (вспомогательные слова), числа (или арифметические выражения) и литералы. Главные и вспомогательные слова операторов постпроцессора приведены в приложениях соответственно. Формат операторов постпроцессора определен в стандарте ISO 3592, ISO 4343 и в методических рекомендациях Госстандарта МР 209—86.

Подмножество операторов постпроцессора, допустимое для определенного типа оборудования с ЧПУ, определяется разработчиком постпроцессора.

Каждому оператору постпроцессора соответствует запись типа 2000 в формате промежуточного языка CLDATA. Вместо главных и вспомогательных слов процессор подставляет их целочисленные коды, вместо арифметических выражений — их значения.

Формат промежуточного файла CLDATA, создаваемого процессором системы Техтран, приведен в приложении 4.

3.2. Начало и конец программы. Вызов постпроцессора

Программа на Техтране *начинается* оператором ДЕТАЛЬ, который имеет формат
ДЕТАЛЬ *'имя детали'*
Оператор конца программы имеет формат
КОНЕЦ
По этому оператору формируются завершающие кадры управляющей программы.
Оператор вызова постпроцессора имеет следующие форматы:
СТАНОК *'имя'*, *a*
СТАНОК *'имя'*, *a* [, p_1 [, p_2]]
где *имя* — кодированное название постпроцессора для соответствующего станка; *a* идентифицирует номер паспорта станка; p_1, p_2 — параметры вызова постпроцессора.

Оператор СТАНОК должен стоять за оператором ДЕТАЛЬ. Он настраивает постпроцессор на конкретную комбинацию станок/устройство ЧПУ.

Значения параметров оператора определяются разработчиком постпроцессора. Так, в постпроцессоре для системы ЧПУ Н-22 оператор СТАНОК '16К20Ф3', 109,1 задает код станка, номер паспорта и первый диапазон кодов оборотов шпинделя, а в постпроцессоре для системы ЧПУ 2С85 оператор СТАНОК 'ЛФ260МФ3', 740,3 идентифицирует паспорт и задает номер программы 3.

3.3. Операторы описания инструмента и управления его сменой

Эта группа операторов предназначена для описания инструмента, поиска инструмента в магазине и загрузки инструмента. Предварительно инструмент должен быть идентифицирован с помощью оператора НОМИНСТР. Это описательный оператор, который не порождает команд на управляющей ленте. Для большей наглядности рекомендуется все операторы НОМИНСТР размещать в начале программы за оператором СТАНОК. Если станок имеет магазин инструментов, то предварительный поиск инструмента задается оператором ВЫБОРИН. Загрузка найденного инструмента (или поворот револьверной головки) осуществляется с помощью оператора ЗАГРУЗ. Для возврата инструмента в магазин предназначен оператор РАЗГРУЗ. Оператор ПРАВКА применяется для придания необходимой формы шлифовальному кругу в профилешлифовальных станках.

Операторы этой группы могут использоваться и для задания коррекции. В этом случае в операторе НОМИНСТР задаются номера корректоров, связанных с данным инструментом. Эти корректоры могут быть заменены операторами ВЫБОРИН и ЗАГРУЗ, если в них задан параметр НОМКОР, или оператором КОРНОМ (см. операторы коррекции). Ввод корректоров, привязанных к инструменту, производится оператором

КОРРЕК $\left\{\begin{matrix}\text{ВКЛ}\\ \text{СПРАВА}\\ \text{СЛЕВА}\end{matrix}\right\}$

В постпроцессор может быть заложена возможность автоматического ввода корректоров по оператору ЗАГРУЗ. В этом случае оператор КОРРЕК следует задавать только тогда, когда необходимо ввести добавочные корректоры, не указанные в операторах управления инструментом.

Оператор задания параметров режущего инструмента объявляет системе имя инструмента, его позицию в магазине и вспомогательном магазине (если он имеется), а также параметры инструмента. Каждый инструмент до его использования в программе должен быть описан оператором НОМИНСТР, т. е. оператор НОМИНСТР *a*, ... должен стоять перед оператором ВЫБОРИН, *a*, ... и ЗАГРУЗ *a*, ... Оператор имеет следующие форматы:

НОМИНСТР *a* [, *b*] [,*c*] $\left\{\begin{matrix}\text{[ВЫЛЕТ, } x, y \text{ [, } z\text{]]}\\ \text{[ДЛИНА, } e\text{]} \left\{\begin{matrix}\text{[, РАДИУС, } r\text{]}\\ \text{[, ДИАМЕТР, } d\text{]}\end{matrix}\right\}\end{matrix}\right\}$

НОМИНСТР a [, b] [, ВЫЛЕТ, x, y [, z]

НОМИНСТР a [, b] [, ДЛИНА, e { [, РАДИУС, r] / [, ДИАМЕТР, d] }]

[, НОМКОР { , [ХКООРД, e] [, УКООРД, f] [, ЗКООРД, g] / , [ХУПЛ, e] [, УЗПЛ, f] [, ЗХПЛ, g] / , [РАДИУС, e] [, ДЛИНА, f] / [i, [, k]] }]

где a задает идентификационный номер инструмента в программе, который указывается в операторах ВЫБОРИН и ЗАГРУЗ; b задает позицию инструмента в магазине; если параметр опущен, то предполагается, что позиция инструмента в магазине совпадает с номером инструмента в программе; c задает позицию инструмента во вспомогательном магазине, если он имеется; ВЫЛЕТ, x, y, z задают приращения x, y, z от настроечной точки инструмента до базовой точки резцедержателя в его рабочем положении с учетом угла установки (или иначе — вылета инструмента по осям координат); в токарных станках параметр ВЫЛЕТ, x, y можно применять для работы с разными режущими кромками одного и того же инструмента, для этого нужно задать в программе несколько операторов НОМИНСТР ... с различными номерами инструмента в программе (a), но с одинаковой позицией в резцедержателе (b) и с различными значениями вылетов; тогда при загрузке того или иного инструмента (b) постпроцессор будет автоматически учитывать смещение данной режущей кромки; ДЛИНА, e задает длину инструмента; РАДИУС, r задает радиус инструмента; ДИАМЕТР, d задает диаметр инструмента; НОМКОР, ... задает номера корректоров; ХКООРД, e, УКООРД, f, ЗКООРД, g включают корректоры по осям координат x, y, z с номерами e, f, g соответственно; ДЛИНА, f задает коррекцию длины инструмента и номер корректора a, используемого для коррекции длины; РАДИУС, e задает коррекцию радиуса режущего инструмента и номер корректора e, используемого для коррекции радиуса; ХУПЛ, e, УЗПЛ, f, ЗХПЛ, g задают плоскость компенсации, в которой действуют корректоры e, f, g соответственно; i, k задают номера корректоров.

Оператор предварительного выбора инструмента для последующей загрузки позволяет совместить поиск инструмента в магазине с обработкой. Оператор имеет следующие форматы:

ВЫБОРИН a

ВЫБОРИН a [, РУЧН] [, НОМКОР, ...]

где a задает идентификационный номер инструмента; РУЧН означает, что выбор инструмента должен производиться вручную, т. е. этим постпроцессору сообщается, что инструмент уже подготовлен к загрузке и никаких действий производить не нужно; НОМКОР, ... — за вспомогательным словом НОМКОР могут следовать те же параметры, что и в операторе НОМИНСТР. Эти параметры позволяют изменить корректоры, связанные с данным инструментом оператором НОМИНСТР.

Оператор загрузки инструмента управляет загрузкой инструмента, описанного ранее оператором НОМИНСТР, и имеет следующие форматы:

ЗАГРУЗ a

ЗАГРУЗ a [, РУЧН] [, НОМКОР, ...]

где a задает идентификационный номер инструмента; РУЧН означает, что загрузка инструмента должна производиться вручную, т. е. этим постпроцессору сообщается, что инструмент уже загружен и никаких действий, связанных с загрузкой инструмента, выполнять не нужно; параметр НОМКОР, ... имеет то же значение, что и в операторе ВЫБОРИН.

Оператор снятия и замены инструмента дает команду на снятие и возврат инструмента в магазин и имеет формат

РАЗГРУЗ

Оператор правки шлифовального круга дает команду на автоматическую правку шлифовального круга в профилешлифовальных станках и имеет формат

ПРАВКА

3.4. Операторы коррекции

В системе Техтран с каждым инструментом могут быть связаны определенные корректоры. Поэтому для многоинструментальных станков операторы этой группы задаются только после загрузки инструмента, т. е. после операторов ВЫБОРИН и ЗАГРУЗ. При этом в операторе КОРРЕК можно не указывать номера корректоров и соответствующие им служебные слова, если эти номера были заданы в операторах управления инструментом.

Оператор КОРНОМ используется для изменения корректоров, связанных с действующим инструментом.

Оператор коррекции размеров и положения инструмента имеет следующие форматы:

КОРРЕК ВКЛ [, ХКООРД, *b*] [, УКООРД, *c*] [, ЗКООРД, *d*]
КОРРЕК ВКЛ, {ХУПЛ, УЗПЛ, ЗХПЛ,}, [НОМКОР,] *e*
КОРРЕК ВКЛ, ДЛИНА, *a*
КОРРЕК ВКЛ *a* [, *b*]
КОРРЕК $\left\{\begin{matrix}\text{СПРАВА}\\ \text{СЛЕВА}\end{matrix}\right\}$, РАДИУС, *f* [, ДЛИНА, *c*]
КОРРЕК ВЫКЛ
КОРРЕК ВЫКЛ [, ХКООРД] [, УКООРД] [, ЗКООРД]
КОРРЕК ВЫКЛ [, ХУПЛ] [, УЗПЛ] [, ЗХПЛ]
КОРРЕК ВЫКЛ, ДЛИНА
КОРРЕК ВЫКЛ, РАДИУС

где *e* — номер корректора для указанной плоскости компенсации; СПРАВА и СЛЕВА задают положение инструмента относительно направления его движения вдоль контура обрабатываемой детали (аналогично операторам ИНСТПР и ИНСТЛВ); ВКЛ применяется при задании коррекции, когда положение инструмента относительно контура обрабатываемой детали безразлично; остальные параметры имеют те же значения, что и в операторе НОМИНСТР (см. с. 00).

Если в операторе отмены коррекции задано только слово ВЫКЛ, то отменяются все введенные корректоры. Для того чтобы отменить нужный корректор, следует после слова ВЫКЛ задать вспомогательное слово, соответствующее отменяемому корректору. Выключать корректоры можно только до загрузки следующего инструмента.

Оператор задания номеров корректоров, связанных с определенным инструментом, может использоваться в постпроцессорах с автоматическим вводом коррекции по оператору загрузки инструмента и имеет формат

КОРНОМ $\left\{\begin{matrix}a\ [, b]\\ \text{ОТМЕН}\ [, b]\end{matrix}\right\}$

где *a* и *b* задают номера корректоров, связанных с действующим инструментом; ОТМЕН задает, что корректор, определяемый параметром *b* или указанный в последнем операторе КОРНОМ, больше не связан с действующим инструментом.

3.5. Операторы управления подачей и шпинделем

Для управления подачей предназначены два оператора: БЫСТРО и ПОДАЧА. Обработка этих операторов зависит от особенностей станка и системы управления, т. е. определяется паспортными данными станка.

Оператор ускоренного движения инструмента означает, что все последующие движения производятся ускоренно до появления оператора ПОДАЧА. Для последующих быстрых перемещений после окончания действия оператора ПОДАЧА необходимо повторить оператор БЫСТРО. Обычно этот оператор применяют для подвода/отвода режущего инструмента, подхода к обрабатываемой поверхности, выхода в начальную точку. Оператор имеет формат

БЫСТРО

Если УЧПУ имеет специальную команду быстрой подачи, то постпроцессор проверяет, является ли заданное направление движения допустимым, и в управляющий

кадр заносится код быстрой подачи. В противном случае выбирается максимальное допустимое для заданного направления значение рабочей подачи.

Если УЧПУ не имеет специальной команды быстрой подачи, то значением подачи становится значение максимальной для данного направления движения подачи из паспорта станка.

Если перемещение в заданном направлении на быстром ходу запрещено, то после отработки текущего кадра на выбранной из паспорта подаче быстрая подача будет восстановлена.

Оператор БЫСТРО нельзя задавать при движении по окружности. В этом случае будет выдано сообщение об ошибке, а в качестве значения рабочей подачи будет выбрано прежнее значение подачи или минимальная подача из паспорта (если рабочая подача не определена). При последующих линейных перемещениях быстрая подача будет восстановлена.

В тех случаях, когда в паспорте станка задано быстрое движение по оператору ИЗ или быстрое движение по оператору ВОЗВРАТ, задавать в программе оператор БЫСТРО перед этими операторами не требуется.

Оператор задания программируемой подачи имеет следующие форматы:

ПОДАЧА *a*

ПОДАЧА $\left\{\begin{matrix}[\text{ММИН},]\ c \\ \text{ММОБ},\ b\end{matrix}\right\}$ [, МАКСММ, *x*]

где *a* задает линейную скорость подачи в мм/мин; ММИН, *c* задает подачу в мм/мин; ММОБ, *b* задает подачу в мм/об; МАКСММ, *x* задает наибольшую допустимую подачу в мм/мин, если подача задана в мм/об.

Запрограммированное значение подачи сохраняется до появления оператора БЫСТРО или нового оператора ПОДАЧА.

Оператор обрабатывается так: значение подачи проверяется на допустимость и, если подача меньше минимальной, то выбирается паспортное значение минимальной подачи, а если подача слишком велика, то выбирается максимальное допустимое для данного направления перемещения значение подачи. Существуют два режима проверки подачи, которые задаются в паспорте. В первом случае должны быть заданы максимальные значения подач при движении по трем, двум и одной координате. Постпроцессор определяет, по какому числу координат задано движение, сравнивает запрограммированное значение подачи с соответствующим значением из паспорта и при необходимости выбирает значение из паспорта. Во втором случае в паспорте задаются максимальные значения подач для каждой оси. Постпроцессор вычисляет проекции вектора подачи на оси координат и сравнивает их с паспортными значениями. Если одна из компонент вектора подачи превышает допустимое значение, то выбирается соответствующее значение из паспорта и подача корректируется. Первый метод проверки подач используется для станков, которые имеют одинаковые приводы по осям *x*, *y* и *z*. Второй метод целесообразно использовать для станков, приводы которых имеют разные характеристики.

Если подача не задана, то выдается сообщение об ошибке и в качестве значения рабочей подачи выбирается значение минимальной подачи из паспорта. При задании подачи в мм/об. предварительно должна быть определена частота вращения шпинделя.

Для управления шпинделем предназначены операторы ШПИНДЛ, ГОЛОВ, ПОВШП, СИНХР.

Оператор задания оборотов шпинделя имеет следующие форматы:

ШПИНДЛ $\left\{\begin{matrix}a \\ \text{ОБМИН},\ a \\ \text{ПОВММ},\ b\end{matrix}\right\}$ $\left[\left\{\begin{matrix}, \text{ПОЧС} \\ , \text{ПРЧС}\end{matrix}\right\}\right]$

ШПИНДЛ $\left\{\begin{matrix}a \\ \text{ОБМИН},\ a \\ \text{ПОВММ},\ b\end{matrix}\right\}$ $\left[\left\{\begin{matrix}, \text{ПОЧС} \\ , \text{ПРЧС}\end{matrix}\right\}\right]\left[\left\{\begin{matrix}\text{ДИАП},\ c \\ [\text{ДИАП},]\end{matrix}\right.\left\{\begin{matrix}\text{НИЖН} \\ \text{СРЕДН} \\ \text{ВЕРХН}\end{matrix}\right\}\right]$

ШПИНДЛ $\left\{\begin{matrix}a \\ \text{ОБМИН},\ a \\ \text{ПОВММ},\ b\end{matrix}\right\}$ $\left[\left\{\begin{matrix}, \text{ПОЧС} \\ , \text{ПРЧС}\end{matrix}\right\}\right]\left[\left\{\begin{matrix}, \text{ДИАП},\ c \\ , [\text{ДИАП},]\end{matrix}\right.\left\{\begin{matrix}\text{НИЖН} \\ \text{СРЕДН} \\ \text{ВЕРХН}\end{matrix}\right\}\right]$

$$\left[\left\{\begin{array}{l}\text{, МАКСММ, } d\\ \text{, МАКСОБМ, } e\end{array}\right\}\right]\left[\left\{\begin{array}{l}\text{, ХКООРД}\\ \text{, УКООРД}\\ \text{, ЗКООРД}\\ \text{, РАДИУС [, } f\text{]}\end{array}\right\}\right]$$

$$\text{ШПИНДЛ}\left\{\begin{array}{l}\text{ВКЛ}\\ \text{ВЫКЛ}\end{array}\right\}$$

$$\text{ШПИНДЛ}\left\{\begin{array}{l}\text{ФИКС}\\ \text{НЕЙТРЛ}\\ \text{ОРИЕНТ}\end{array}\right\}$$

где ОБМИН, *a* задает частоту вращения шпинделя в об/мин; ПОВММ, *b* задает скорость резания в мм/мин; система может поддерживать постоянную скорость резания, если задан этот параметр; пересчет числа оборотов шпинделя будет выполняться для каждого участка траектории (отрезка прямой или дуги окружности). Максимальная частота вращения (в об/мин) может быть задана параметром МАКСОБМ; МАКСММ, *d* задает максимальную скорость резания для заданного значения числа оборотов шпинделя; МАКСОБМ, *e* задает максимальное значение числа оборотов шпинделя для заданной скорости резания; ХКООРД, УКООРД, ЗКООРД указывают, что текущее значение данной координаты будет использоваться при расчете скорости резания; РАДИУС, *f* задает радиус, который используется при расчете скорости резания, если задано ПОВММ; ВКЛ включает шпиндель или восстанавливает скорость шпинделя после операторов ТЕХОСТ или СТОП; ВЫКЛ выключает шпиндель; ФИКС задает зажим шпинделя; НЕЙТРЛ задает отключение привода шпинделя; ОРИЕНТ задает поворот шпинделя до фиксированного положения и его зажим.

Оператор ШПИНДЛ должен обязательно появиться перед любым рабочим движением станка. Он действует до появления нового оператора ШПИНДЛ, операторов ТЕХОСТ или СТОП.

Оператор выбора определенной шпиндельной головки имеет формат

ГОЛОВ *a*
ГОЛОВ ОБА
ГОЛОВ ГЛАВН
ГОЛОВ {ВЕРХН | НИЖН | СЛЕВА | СПРАВА}
ГОЛОВ {ПИНОЛЬ | СУППОРТ}

где *a* — номер шпиндельной головки; ОБА задает обе головки (обычно требуется синхронизация); ГЛАВН задает главную головку; ВЕРХН задает верхнюю головку; НИЖН — нижнюю головку; СЛЕВА — левую головку; СПРАВА — правую головку; ПИНОЛЬ задает соответствующую пиноль шпинделя и СУППОРТ — соответствующую головку суппорта.

Оператор поворота шпиндельной головки позволяет повернуть шпиндельную головку в определенную позицию или на заданный угол и имеет формат

$$\text{ПОВШП}\left\{\begin{array}{l}a\\ \text{ПРИРАЩ, } b\end{array}\right\}\left[\left\{\begin{array}{l}\text{, ПОЧС}\\ \text{, ПРЧС}\end{array}\right\}\right]\text{[, ФИКС]}\left[\left\{\begin{array}{l}\text{, ТЕКУЩ}\\ \text{, СЛЕД}\end{array}\right\}\right]$$

где *a* задает позиционирование шпиндельной головки в угол *a* в градусах, если определено положение абсолютного нуля. Отрицательное значение задает поворот по часовой стрелке, положительное — против часовой стрелки; ПРИРАЩ, *b* задает поворот шпиндельной головки на угол в градусах, иначе — приращение угла в градусах, где отрицательное направление вращения — по часовой стрелке, положительное — против часовой стрелки; приращение берется относительно предыдущего положения шпиндельной головки; ПОЧС указывает на отрицательное направление вращения (по часовой стрелке); ПРЧС указывает на положительное направление вращения (против часовой стрелки); ФИКС задает фиксацию шпиндельной головки; ТЕКУЩ указывает, что команда на поворот должна быть выдана в текущем кадре; СЛЕД указывает, что команда на поворот должна быть выдана в следующем кадре.

Оператор синхронизации служит для синхронизации различных рабочих движений станка (синхронизация подачи и вращения шпинделя при резьбонарезании, синхронизация движений суппортов для станков с несколькими суппортами и др.).

Оператор имеет формат

СИНХР { ВКЛ / ВЫКЛ }

где ВКЛ и ВЫКЛ — начало и конец синхронизации.

3.6. Операторы отвода и возврата инструмента

Эта группа операторов используется для возврата инструмента в исходную позицию (ВОЗВРАТ), отвода инструмента в плоскость безопасности (БЕЗОПЛ, ОТВОД), задания безопасного расстояния, на которое отводится инструмент в циклах (БЕЗОПРСТ) и для задания точки смены инструмента (ТЧКСМЕН).

Оператор задания безопасной плоскости для отвода инструмента позволяет задать плоскость, в которую отводится режущая кромка инструмента по оператору ОТВОД. В качестве безопасной плоскости обычно используют плоскость, параллельную рабочей плоскости. Оператор имеет формат

БЕЗОПЛ { ХУПЛ / УЗПЛ / ЗХПЛ }, *a*

где *a* задает расстояние между плоскостью безопасности и соответствующей координатной плоскостью; ХУПЛ, УЗПЛ, ЗХПЛ задают координатную плоскость *xy*, *yz* и *zx* соответственно.

Оператор задания безопасного расстояния используют для задания отвода инструмента при программировании стандартных циклов. Он имеет формат

БЕЗОПРСТ *a*

где *a* — безопасное расстояние от поверхности детали, на которое нужно отводить инструмент, чтобы избежать столкновения его с деталью при перемещении из одной точки в другую.

Оператор отвода инструмента в безопасную плоскость имеет формат

ОТВОД

Если система ЧПУ имеет специальную команду для отвода инструмента, то по оператору ОТВОД генерируется соответствующий код. Если же таким свойством система ЧПУ не обладает, то по оператору ОТВОД генерируется перемещение в плоскость безопасности, заданную оператором БЕЗОПЛ, по нормали к рабочей плоскости.

Оператор возврата инструмента в исходное положение имеет формат

ВОЗВРАТ

Если станок имеет стандартную позицию возврата, то по оператору ВОЗВРАТ формируется требуемый кадр (или группа кадров). Если же стандартной позиции возврата станок не имеет, то по этому оператору инструмент будет перемещен по кратчайшему пути в точку, заданную оператором ИЗ, или в точку, определенную в операторе ТЧКСМЕН.

Оператор указания безопасной позиции для смены инструмента задает высоту *h* от плоскости обрабатываемой детали, на которую следует отвести настроечную точку инструмента при его смене (задается в системе координат детали) либо точку смены инструмента. Оператор имеет формат

ТЧКСМЕН *h*

ТЧКСМЕН *x*, *y* [, *z*]

ТЧКСМЕН ОТМЕН

где *h* — высота точки смены инструмента от поверхности детали; *x*, *y*, *z* — координаты точки, в которую следует отвести инструмент при его смене (по оператору ВОЗВРАТ); ОТМЕН указывает, что заданная ранее позиция смены инструмента отменяется и восстанавливается исходная позиция возврата.

3.7. Операторы управления формированием перфоленты

Оператор длины лидера задает длину заправочной части перфоленты в сантиметрах и имеет формат

ЛИДЕР *a*

Оператор изменения нумерации кадров управляющей программы применяют, если есть необходимость в специальной нумерации кадров. Он позволяет изменять порядковый номер следующего кадра и шаг нумерации. По умолчанию кадры управляющей программы нумеруются с единицы (шаг также равен единице). Оператор имеет формат

ПОРНОМ $\left\{\begin{matrix} a \\ a, \text{ ПРИРАЩ}, b \\ \text{ПРИРАЩ}, b \end{matrix}\right\}$

где *a* задает номер следующего кадра; шаг нумерации равен 1; *a*, ПРИРАЩ, *b* указывает, что следующий кадр управляющей программы получит номер *a*, а последующие кадры пронумерованы с шагом *b*; ПРИРАЩ, *b* задает новый шаг нумерации.

Различные операторы ПОРНОМ являются модальными, т. е. каждое условие остается в силе, пока не будет отменено другим оператором ПОРНОМ.

Оператор создания маркера на перфоленте имеет следующие форматы:

МАРКЕР

МАРКЕР *a*

МАРКЕР $\left\{\begin{matrix} \text{ВКЛ} \\ \text{ВЫКЛ} \end{matrix}\right\}$

где *a* задает идентификатор метки на ленте.

Маркер, заданный на перфоленте, используется либо для возврата к нему по оператору ПЕРЕМТК, либо для поиска определенного участка ленты. Оператор МАРКЕР без параметров используется для маркировки главного кадра символом «:».

Оператор перемотки задает перемотку управляющей ленты до символа *a*, определенного как идентификатор метки оператором МАРКЕР, либо к началу ленты. Оператор имеет следующие форматы:

ПЕРЕМТК *a*

ПЕРЕМТК $\left\{\begin{matrix} \text{ВКЛ} \\ \text{ВЫКЛ} \end{matrix}\right\}$

где *a* задает символ, до которого перематывается лента; ВКЛ и ВЫКЛ включает и отключает перемотку ленты по оператору КОНЕЦ соответственно.

Оператор пропуска дает возможность пропускать помеченные кадры управляющей программы и имеет формат

ПРОПБЛ $\left\{\begin{matrix} \text{ВКЛ} \\ \text{ВЫКЛ} \end{matrix}\right\}$

где ВКЛ и ВЫКЛ указывают начало и конец информации для условного пропуска.

Оператор ПРОПБЛ вызывает маркировку всех последующих кадров символом «/» в начале кадра. Эти кадры обрабатываются устройством управления станка, если на пульте включен тумблер пропуска кадров.

Оператор конца замкнутого цикла обработки имеет формат

АБЗАЦ

Если после оператора АБЗАЦ процесс обработки продолжается, то выводится новая управляющая лента (вместе с новым заголовком).

Оператор разбиения программы указывает место разбиения управляющей программы, если изготовить единый программоноситель по каким-либо причинам невозможно (например, из-за ограничения по длине). Оператор имеет формат

РАЗБПР

Оператор вставки символов непосредственно в управляющую программу позволяет выводить символы (литералы), образующие команды устройства ЧПУ, непосредственно на программоноситель и имеет формат

ВСТАВ *литерал*, ТЕКУЩ

Литерал вставляется в конец текущего кадра. Если литерал должен завершить формирование кадра, то следует задать ТЕКУЩ, иначе информация, сформированная по следующим операторам постпроцессора, будет добавлена вслед за литералом. Если требуется вывести строку символов отдельным кадром, то следует задать

ППФУН ′ ТЕКУЩ ′

ВСТАВ ′ . . . ′, ТЕКУЩ

Сокращенная форма записи оператора имеет вид:

литерал$_1$

литерал$_2$

. . . . .

Каждый литерал будет выведен отдельным кадром. Это позволяет вставлять в программу на Техтране фрагменты программы в кодах системы ЧПУ. Пользоваться оператором ВСТАВ следует с осторожностью, так как если в операторе заданы коды функций M и G, то состояние станка после их выполнения не будет соответствовать состоянию постпроцессора.

Оператор печати текста в выходном документе постпроцессора имеет формат

ППЕЧ *литерал*

Если литерал начинается не со скобки, то он печатается в выходном документе и заносится в файл инструкций оператору станка, например: ППЕЧ 'НАЧАЛО ОБРАБОТКИ'.

Если литерал начинается со скобки, то весь текст вместе со скобками печатается, вносится в инструкцию, а также вносится в конец текущего кадра. Этот формат позволяет выводить на программоноситель (перфоленту) комментарии, пропускаемые системой управления.

При выводе комментариев на программоноситель в их тексте не должно быть символов русского алфавита.

3.8. Стандартные циклы

Для задания параметров стандартных циклов, исполняемых системой ЧПУ, предназначен оператор ЦИКЛ. Он задает выполнение определенной последовательности действий в каждой точке траектории. Оператор *имеет* следующие форматы:

Формат 1

ЦИКЛ { ВКЛ / ВЫКЛ / ОТМЕН / РУЧН }

Формат 2

ЦИКЛ { СВЕРЛ / ТОРЦОВ / РЕЗБМЕТ / РАЗВЕР / РАСТОЧ / РАСТОЧ5 / РАСТОЧ6 / РАСТОЧ7 / РАСТОЧ8 / РАСТОЧ9 / РЕЗБМЕТ1 }, z, { ММИН, d / ММОБ, e }, f

Формат 3

ЦИКЛ ГЛУБОК, z, ПРИРАЩ, q, { ММИН, d / ММОБ, e }, f

Формат 4

ЦИКЛ КЗЕНК, k, l, { ММИН, d / ММОБ, e }, f

В операторе ЦИКЛ 1-го формата используются следующие параметры: ВЫКЛ, ОТМЕН отключают выполнение в каждой точке последовательности действий, заданной предшествующим оператором ЦИКЛ; ВКЛ возобновляет последовательность операций, отключенную оператором ЦИКЛ ВКЛ; РУЧН останавливает инструмент в каждой точке траектории (для того чтобы оператор мог выполнить незапрограммированные действия).

Оператор ЦИКЛ 2-го формата вызывает последовательность действий, соответствующую встроенным циклам G81—G89 в формате ISO: СВЕРЛ — сверление (G81); ТОРЦОВ — подрезка торцов (G82); РЕЗБМЕТ — нарезание резьбы метчиком (G84); РАСТОЧ5 или РАСТОЧ — расточка (G85); РАСТОЧ6 или РАЗВЕР — развертывание (G86); РАСТОЧ7 — расточка (G87); РАСТОЧ8 — расточка (G88); РАСТОЧ9 — расточка (G89); РЕЗБМЕТ1 — нарезание резьбы метчиком (G63); z — итоговая глубина обработки (от поверхности детали); d — подача в мм/мин; e — подача в мм/об; f — высота (над поверхностью детали), до которой происходит быстрое позиционирование до начала обработки и выход после ее окончания; из этой точки начинается движение на рабочей подаче.

Оператор ЦИКЛ 3-го формата используется для задания цикла глубокого сверления (G83). Параметры z, d, e и f описаны выше; q — приращение глубины опускания инструмента на каждом шаге обработки.

Оператор ЦИКЛ 4-го формата определяет цикл зенкерования (G82). Параметры k и l означают диаметр зенкера на итоговой глубине и рабочий угол зенкера соответственно.

Глубину обработки вычисляют по формуле:

$$z=\frac{k}{2\,\mathrm{tg}\,(l/2)}.$$

Высота над поверхностью детали, на которой происходит быстрое позиционирование при перемещении из одной точки в другую и на которую выводится инструмент по окончании каждого цикла (рис. 3.1), должна задаваться оператором БЕЗОПРСТ до включения цикла (a — безопасное расстояние; z — глубина отверстия; f — недоход до поверхности детали; r — перемещение на быстром ходу; z_1 — перемещение на рабочем ходу; *1* — запрограммированная точка; *2* — начало и конец стандартного цикла), например:

1) оператор ЦИКЛ СВЕРЛ, 30, ММИН, 10,1 задает цикл сверления на глубину 30 мм на рабочей подаче 10 мм/мин; недоход до материала — 1 мм;

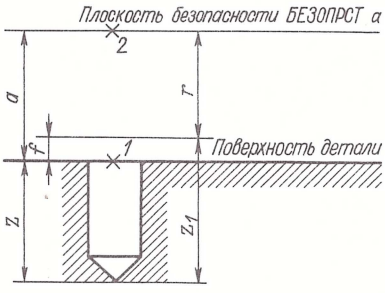


Рис. 3.1

2) оператор ЦИКЛ ГЛУБОК, 70, ПРИРАЩ, 15, ММИН, 10, 1,5 задает цикл глубокого сверления на глубину 70 мм с шагом 15 мм на рабочей подаче 10 мм/мин; недоход до материала — 1,5 мм;

3) следующая последовательность операторов задает выполнение цикла в трех точках:

```
ИЗ 0,0
БЫСТРО
ЦИКЛ СВЕРЛ, 30, ММИН, 10, 1
БЕЗОПРСТ 20
ВТОЧКУ 10,10 : 20,20: 30,30
ЦИКЛ ВЫКЛ
ВТОЧКУ 0,0
```

3.9. Прочие операторы

Оператор управления включением и типом охлаждения имеет следующие форматы:

ОХЛАД { ВКЛ | ВЫКЛ }

ОХЛАД { ЖИДК | ТУМАН }

ОХЛАД a_1 [, a_2] [, a_3] ...

где ВКЛ и ВЫКЛ включают и выключают охлаждение соответственно; ЖИДК и ТУМАН включают подачу охлаждения жидкостью и туманом соответственно; a_1, a_2, a_3, ... — номера трубопроводов, подающих охлаждающий компонент.

Оператор вставки кодов подготовительных функций (*G*) имеет формат

ПДГФУН a_1, a_2, a_3, . . . [, ТЕКУЩ]

где a_1, a_2, a_3 — кодовые номера вспомогательных функций типа М, которые должны быть внесены в текущий кадр; ТЕКУЩ завершает формирование текущего кадра.

При использовании этого и следующего операторов следует соблюдать осторожность, так как состояние постпроцессора не будет соответствовать состоянию станка, измененному командой G (или М).

Оператор вставки кодов вспомогательных функций (М) имеет формат

ВСПФУН a_1, a_2, a_3, . . . [, ТЕКУЩ]

где a_1, a_2, a_3 — кодовые номера вспомогательных функций типа М, которые должны быть внесены в текущий кадр; ТЕКУЩ завершает формирование текущего кадра.

Оператор управления поворотным столом позволяет повернуть стол в определенную позицию или на заданный угол и имеет формат

$$\text{СТОЛ} \left\{ \begin{matrix} a \\ \text{ПРИРАЩ, } b \end{matrix} \right\} \left[\left\{ \begin{matrix} \text{, ПОЧС} \\ \text{, ПРЧС} \end{matrix} \right\} \right] \text{ [, ФИКС] [, ТЕКУЩ]}$$

где a задает позиционирование стола в угол a в градусах, если определено положение абсолютного нуля; ПРИРАЩ, b задает поворот стола на угол b в градусах, иначе — приращение угла в градусах; ФИКС задает фиксацию поворотного стола; ТЕКУЩ указывает, что команда на поворот стола должна быть выдана в текущем кадре.

Оператор системы координат станка позволяет определить координаты центра поворотного стола в системе координат детали и имеет формат

КООРДСТ x, y [,z]

где x, y — координаты стола в системе координат детали; z — координата установочной поверхности.

Оператор останова станка и считывающего устройства имеет формат

СТОП

При последующем пуске станка работа будет продолжена. Оператор используют в том случае, если в процессе обработки требуется остановка (например, для контрольного замера детали).

Оператор технологического останова станка позволяет задать останов станка и считывающего устройства включением соответствующего тумблера на пульте ЧПУ и имеет формат

ТЕХОСТ

Оператор паузы в работе станка имеет формат

$$\text{ПАУЗА} \left\{ \begin{matrix} a \\ \text{ОБОРОТ, } b \end{matrix} \right\}$$

где a задает продолжительность паузы (с); ОБОРОТ, b задает продолжительность паузы числом оборотов шпинделя; в этом случае предварительно должна быть задана частота вращения шпинделя.

Оператор переноса начала системы координат станка

СДВИГ x, y [, z]

Оператор позволяет осуществить алгебраическое преобразование координат. Параметры x, y, z задают начало новой системы координат станка в системе координат детали. Повторные операторы СДВИГ заменяют предыдущий оператор.

Оператор задания шага резьбы

ШАГРЕЗ a [, b]

где a и b — шаг резьбы по продольной и поперечной (для конических разьб) координатам соответственно.

Оператор ШАГРЕЗ позволяет задать однопроходный цикл резьбонарезания (типа G33 в УЧПУ Н-22), например:

```
ВТОЧКУ ТЧ1
ШАГРЕЗ 1.5
ВТОЧКУ ТЧ2
```

Для нарезания резьб за несколько проходов может быть использован макрос REZXBA, имеющийся в составе библиотеки:

ВЫЗОВ РЕЗЬБА (ШР=*число*, НИТ=*число*, ЧЕРПР=*число*, %
(ТЧНР=*точка*), (ТЧКР=*точка*), (ВН=*число*)

где ШР — шаг резьбы (мм); НИТ — число заходов резьбы; ЧЕРПР — число черновых проходов (число чистовых проходов равно 2, что соответствует 0,02 мм); ТЧНР — точка, определяющая начало резьбы на заготовке. Ее координата x задает начало резьбы по чертежу, а координата y — наружный радиус резьбы; ТЧКР — точка, определяющая конец резьбы; ВН — тип резьбы (наружная резьба +1, внутренняя резьба −1).

Оператор задания режима подачи смеси для газовой резки позволяет включать (ВКЛ) и выключать (ВЫКЛ) газорезательный автомат и имеет формат

РЕЗКА { ВКЛ | ВЫКЛ }

Оператор выбора рабочей плоскости

РЕЖИМ ПЛОСК, {ХУПЛ|УЗПЛ|ЗХПЛ|ОТМЕН}

где ХУПЛ, УЗПЛ, ЗХПЛ задают требуемую рабочую плоскость (по умолчанию — ХУПЛ); ОТМЕН задает возврат к исходной рабочей плоскости.

Оператор скругления траектории задает скругление траектории движения центра режущего инструмента и имеет формат

РЕЖИМ РАДИУС, { r | ОТМЕН }

где РАДИУС, r задает радиус скругляющей окружности $R_{скр}$, который вычисляется следующим образом: 1) если $0<r<R_и$, то $R_{скр}=r$ при обходе контура изнутри и $R_{скр}=R_и-r$ при обходе контура снаружи; 2) если $r=0$, то скругление выполняется радиусом $R_{скр}=0{,}1R_и$ при обходе контура изнутри и $R_{скр}=0{,}9R_и$ — снаружи; РАДИУС, ОТМЕН выключает скругление траектории ($R_и$ — радиус инструмента).

Вставка скругляющих дуг начинается за оператором движения, перед которым задан оператор коррекции радиуса:

КОРРЕК { СПРАВА | СЛЕВА }

Оператор специальных функций постпроцессора определяет действия, задаваемые технологом-программистом для управления постпроцессором и имеет формат

ППФУН { *литерал* | *служебное слово* | *число* } ...

где *литерал*, *служебное слово*, *число* — последовательность элементов (не более десяти), обеспечивающих инициализацию или изменение внутренних параметров постпроцессора. Применение этого оператора регламентируется разработчиком постпроцессора.

Так, оператор

ППФУН { 'АБС' | 'ПРИРАЩ' }

используется для выбора режима работы постпроцессора — в абсолютных значениях или приращениях.

Для работы с подпрограммами предназначены следующие операторы:

ППФУН 'ППНАЧ', *имя подпрогр*

— задает начало подпрограммы ('ППНАЧ' означает, что, начиная с этого оператора, вся информация на управляющей перфоленте оформляется в виде подпрограммы; *имя подпрогр* — литерал или число, идентифицирующие подпрограмму;

ППФУН 'ППКОН'

— означает, что заканчивается оформление информации на управляющей ленте в виде подпрограммы;

ППФУН 'ВЫЗОВ', *имя подпрогр*, *а*

где *а* — число повторений;

ППФУН 'ПАРАМ', *имя параметра*$_1$, a_1, *имя параметра*$_2$, a_2, . . .

где *имя параметра*$_1$, . . . — литерал (или число), идентифицирующие данный параметр; a_1, . . . — значение, присваиваемое параметру.

Оператор используют для формирования списка формальных параметров и присваивания им фактических значений. В одном операторе разрешается определить не более четырех параметров.

3.10. Примеры использования операторов постпроцессора в программах

В примере, приведенном ниже, показана типичная последовательность операторов постпроцессора, необходимых при программировании обработки на многоинструментальном станке:

```
ДЕТАЛЬ 'ПРОГРАММА1'
СТАНОК '2206 ВМФ4', 720
# ОПИСАНИЯ ИНСТРУМЕНТОВ
НОМИНСТР 1,10, ДЛИНА, 200, НОМКОР, ХКООРД, 1, УКООРД, 2, ЗКООРД, 3
НОМИНСТР 2,7, ДЛИНА, 150, НОМКОР, ХКООРД, 11, УКООРД, 12,3, ЗКООРД, 13
НОМИНСТР 3,20, ДЛИНА, 145
ИЗ 0,0                    # НАЧАЛО ДВИЖЕНИЯ
ВЫБОРИН 1                 # ВЫБОР 1-го ИНСТРУМЕНТА
ЗАГРУЗ 1                  # ЗАГРУЗКА 1-ГО ИНСТРУМЕНТА
КОРРЕК ВКЛ                # ВВОД КОРРЕКЦИИ ПОЛОЖЕНИЯ
БЫСТРО                    # БЫСТРАЯ ПОДАЧА
ВТОЧКУ ТЧ1
ПОДАЧА 100                # ПОДАЧА 100 ММ/МИН
ОХЛАД ВКЛ                 # ВКЛЮЧЕНИЕ ОХЛАЖДЕНИЯ
ШПИНДЛ 600                # ВКЛЮЧЕНИЕ ШПИНДЕЛЯ ПО ЧАС. СТРЕЛКЕ
КОРРЕК РАДИУС, 6          # ВВОД КОРРЕКЦИИ РАДИУСА
ИДИ ДО ПР1
. . . . . .
ВЫБОРИН 2                 # ВЫБОР 2-ГО ИНСТРУМЕНТА
. . . . . .
КОРРЕК ВЫКЛ, РАДИУС       # ОТКЛЮЧЕНИЕ КОРРЕКЦИИ РАДИУСА
. . . . . .
КОРРЕК ВЫКЛ               # ОТКЛЮЧЕНИЕ ВСЕХ КОРРЕКТОРОВ
ВОЗВРАТ                   # ВОЗВРАТ В ИСХОДНУЮ ПОЗИЦИЮ
ЗАГРУЗ 2                  # ЗАГРУЗКА 2-ГО ИНСТРУМЕНТА
ПОДАЧА 200                # ПОДАЧА 200 ММ/МИН
ШПИНДЛ 1200               # ОБОРОТЫ ШПИНДЕЛЯ
. . . . . .
ВЫБОРИН 3                 # ВЫБОР 3-ГО ИНСТРУМЕНТА
. . . . . .
ЗАГРУЗ 3                  # ЗАГРУЗКА 3-ГО ИНСТРУМЕНТА
ПОДАЧА ММИН, 50           # ПОДАЧА 50 ММ/МИН
ШПИНДЛ 800                # ОБОРОТЫ ШПИНДЕЛЯ
СТОЛ 15, ПОЧС             # ПОВОРОТ СТОЛА НА 15° ПО ЧАСОВОЙ СТРЕЛКЕ
. . . . . .
СТОП
КОНЕЦ
```

В следующем примере показано использование операторов постпроцессора при программировании токарной обработки:

```
ДЕТАЛЬ 'ПРОГРАММА 2'
СТАНОК '16К20Т1', 140
# ОПИСАНИЯ ИНСТРУМЕНТОВ
НОМИНСТР 1,1, ВЫЛЕТ, —15, —100
НОМИНСТР 2,1, ВЫЛЕТ, 15, —100
НОМИНСТР 3,2, ВЫЛЕТ, —70, —110
. . . . . .
```

```
ИЗ 0,0                                  # НАЧАЛО ДВИЖЕНИЯ
ЗАГРУЗ 1                                # ЗАГРУЗКА ПЕРВОГО ИНСТРУМЕНТА
ВТОЧКУ ТЧ1
ПОДАЧА ММИН, 50                         # ПОДАЧА 50 ММ/МИН
ШПИНДЛ ПОВММ, 5000, ПОЧС, ДИАП,1,%      # СКОРОСТЬ РЕЗАНИЯ
        МАКСОБМ, 1200                   # 5 М/МИН
ОХЛАД ВКЛ
. . . . . .
ЗАГРУЗ 2, РУЧН                          # ВВОД 2-Й РЕЖУЩЕЙ КРОМКИ
ПОДАЧА ММИН, 100                        # ПОДАЧА 100 ММ/МИН
ШПИНДЛ 600, ПОЧС, ДИАП, 1               # ОБОРОТЫ И ДИАПАЗОН
. . . . . .
ЗАГРУЗ 3                                # ЗАГРУЗКА 3-ГО ИНСТРУМЕНТА
ПОДАЧА ММИН, 400                        # ПОДАЧА 400 ММ/МИН
ШПИНДЛ 800, ПОЧС, ДИАП, 2               # ОБОРОТЫ И ДИАПАЗОН
. . . . . .
КОНЕЦ
```

В этой программе оператор ШПИНДЛ используется для задания скорости резания 5 м/мин. Постпроцессор будет поддерживать в ходе обработки заданную скорость резания до прихода нового оператора ШПИНДЛ. Параметр МАКСОБМ задает максимальное значение числа оборотов. Кроме того, следует обратить внимание на то, что в программе описана обработка двумя режущими кромками. Параметр РУЧН в операторе ЗАГРУЗ подавляет формирование команды смены инструмента в управляющей программе.

Глава 4

ТЕХНОЛОГИЧЕСКАЯ ПОДГОТОВКА ПРОИЗВОДСТВА, ПРОГРАММИРОВАНИЕ И КОНТРОЛЬ

4.1. Особенности технологической подготовки производства в условиях автоматизированного расчета управляющих программ

Появление и широкое внедрение оборудования с ЧПУ вызвало качественные изменения как процесса производства, так и процесса технологической подготовки. Автоматизация процесса обработки, концентрация операций, возможность реализации сложных пространственных перемещений инструмента, оптимизация режимов резания, быстрая переналаживаемость оборудования, многостаночное обслуживание — все эти факторы приводят к резкому снижению трудоемкости обработки. В то же время трудоемкость и сроки технологической подготовки увеличиваются, что обусловлено необходимостью особо тщательной проработки технологических решений и появлением нового, достаточно трудоемкого этапа ТПП — подготовки управляющих программ. Эти изменения сопровождаются увеличением объема технологической документации и ее информационной насыщенности. Так, объем информации в технологической документации для оборудования с ЧПУ примерно в 25 раз больше, чем в документации для универсального оборудования, объем выполняемых графических работ возрастает в 20 раз, что приводит к увеличению общей трудоемкости разработки технической документации в 4—5 раз [6].

Эти изменения в ТПП обусловлены тем, что управляющая программа является числовой моделью технологического процесса обработки, для создания которой требуется выполнить преобразование в числовую форму большого количества разнородной информации. В данных условиях естественным и единственно возможным способом повышения оперативности ТПП, сокращения ее трудоемкости и обеспечения информационной, программной и технической (аппаратной) совместимости с системами автоматизации проектирования (САПР) и управления производством (АСУП) является комплексная автоматизация технологического проектирования [3].

На практике решение этой задачи производится ноэтапно, путем внедрения отдельных автоматизированных систем, решающих частные задачи технологического проектирования, а именно: проектирование операционной технологии, группирование деталей по конструкторско-технологическим признакам, поиск, заимствование и проектирование средств технологического оснащения и т. д. Очевидно, что автоматизация отдельного этапа ТПП не может не повлиять на состав и характер остальных работ по технологическому проектированию. Автоматизация подготовки управляющих программ также выдвигает к остальным этапам ТПП специфические требования, зависящие от функциональных возможностей и технических характеристик конкретной САП. В данной главе сделана попытка показать влияние САП Техтран на характер и содержание основных этапов ТПП для оборудования с ЧПУ.

Общие требования и этапы ТПП для оборудования с ЧПУ определены ГОСТ 14.002—73 и ГОСТ 23501.603—84. В частности ГОСТами предусмотрено выполнение следующих работ:

обеспечение технологичности конструкции деталей;

разработка технологических процессов;

проектирование и изготовление средств технологического оснащения;

программирование технологических и вспомогательных операций;

управление процессом ТПП;

внедрение управляющих программ.

Таблица 4.1

Функциональные возможности САП Техтран	Область проявления технологичности	
	Технологическая подготовка	Изготовление
Входной язык:		
система координат	+	—
средства для вычислений	+	—
преобразование траектории и геометрических объектов	+	—
операторы организации циклов	+	—
средства описания сплайнов	+	+
средства описания наклонных поверхностей	+	+
макроаппарат	+	—
Технологический процессор:		
способы описания контура детали и заготовки	+	—
проектирование траектории	+	+
расчет режимов обработки	—	+
база технологических данных	+	—
Постпроцессоры	—	—

Примечание. Знак «+» означает наличие влияния соответствующей функции на область проявления технологичности.

Обеспечение технологичности конструкции деталей. Технологичность конструкции — это совокупность свойств конструкции изделия, обеспечивающих возможность применения более эффективной технологии, технического обслуживания и ремонта при заданных показателях качества и условиях изготовления и эксплуатации. Таким образом, различают производственную и эксплуатационную технологичность. Оценка уровня технологичности конструкции производится по показателям, состав и правила выбора которых регламентированы ГОСТ 14.201—73 и ГОСТ 14.202—73. Укрупненно производственную технологичность можно определить как приспособленность изделия к сокращению затрат на подготовку производства и изготовление изделия. При использовании САП производственную технологичность следует дополнительно оценивать, во-первых, с точки зрения возможности сокращения затрат на ТПП и, во-вторых, с точки зрения допустимости использования возможностей САП, непосредственно влияющих на эффективность процесса обработки.

В табл. 4.1 приведены функциональные возможности САП Техтран, с учетом которых должна выполняться отработка на технологичность.

Прямоугольная система координат, используемая в большинстве САП, должна применяться при простановке размеров. Наличие во входном языке операторов пре-

образования координат допускает при простановке размеров использовать локальные координатные системы с началом в произвольной точке и произвольным направлением осей.

Вычислительные средства, представляемые в языке Техтран в виде массивов, арифметических выражений и стандартных математических функций, делают более предпочтительными аналитический способ задания криволинейного контура и табличное задание координат большого числа точек.

Преобразование траектории и геометрических объектов позволяет сократить объем исходной информации, так как симметричные, повторяющиеся геометрические объекты и траектория их обработки могут быть описаны один раз — только для базового (исходного) контура. Остальные участки обработки получаются при помощи матричных преобразований всего исходного контура или его отдельных участков. При составлении программы необходимо, чтобы в чертеже детали максимально использовалась симметрия, а отдельные конструктивные элементы были унифицированы.

Операторы организации циклов позволяют многократно повторять какой-либо участок программы, что сокращает ее объем и снижает трудоемкость отладки. Эта возможность процессора системы дает наибольший эффект при описании большого числа однотипных геометрических элементов, повторяющихся контуров, которые желательно задавать на чертеже в виде упорядоченных структур точек, прямых, окружностей или объектов типа «КОНТУР».

Средства описания сплайнов позволяют программировать обработку на станках с ЧПУ профилей, заданных набором точек. При ручном способе подготовки УП возможен только графический способ построения эквидистанты, который не дает требуемой точности. При использовании САП, не имеющей специальных средств описания таких кривых, требуется выполнить значительный объем вспомогательных геометрических построений, которые тем не менее не дают достоверных результатов. В обоих случаях конструкция детали не может быть признана технологичной, так как требуется доработка контура на профилешлифовальном станке или по шаблону. При использовании САП Техтран нет ограничений на задание профиля набором точек, что упрощает технологическую проработку, составление УП и сам процесс изготовления.

Средства описания наклонных поверхностей САП Техтран позволяют реализовать контурную обработку с одновременным перемещением по 3-й координате. Благодаря этому в конструкции деталей допускается применять колодцы, карманы и стенки, имеющие наклонное дно. Таким образом, снимается одно из существенных ограничений, обусловленное сложностью и трудоемкостью расчета траектории обработки таких конструктивных элементов, полнее реализуются технологические возможности станков с ЧПУ.

Макроаппарат позволяет разрабатывать макросы — параметризованные программы на входном языке Техтран. Макросы могут быть разработаны для типовых технологических переходов или унифицированных операций обработки типовых деталей. Создание библиотек макросов на типовые технологические переходы и унифицированные операции следует рассматривать как составляющую часть комплекса работ по внедрению группового метода производства. Для разработки типовых программ необходимо провести стандартизацию, унификацию и группирование деталей и их элементов по конструктивным признакам, унификацию технологических процессов и средств оснащения [8].

Технологический процессор, его использование в САП Техтран накладывает следующие ограничения при отработке чертежей на технологичность:

1) контур заготовки и детали должен состоять из прямых и дуг окружностей;

2) форма канавок должна допускать их обработку прямыми канавочными резцами;

3) фаски и скругления контура детали считаются сопрягающими элементами и не могут следовать друг за другом. Угол наклона фаски рекомендуется проставлять от горизонтальной оси, ширину фаски — в направлении той же оси;

4) материал заготовки (если это допустимо по конструктивным соображениям) рекомендуется назначать из числа материалов, имеющихся в технологической базе.

Постпроцессоры обычно не накладывают дополнительных ограничений на конструкцию детали, так как обеспечивают реализацию всех технологических возможностей оборудования с ЧПУ. Деталь является технологичной, если может быть обработана на станке, для которого имеется постпроцессор системы Техтран. Вместе с тем существующие ограничения по числу одновременно управляемых координат не позволяют реализовать некоторые постпроцессорные функции. Например, невозможно запрограммировать разворот шлифовального круга вокруг вертикальной оси или реализовать программируемое отклонение от вертикали проволоки электроискрового станка. Отсутствие в САП Техтран средств описания синхронизированных движений нескольких рабочих органов является препятствием для создания постпроцессоров, полностью реализующих технологические возможности токарно-револьверных автоматов и карусельных станков с ЧПУ. Если ограничения такого рода могут быть учтены при технологическом контроле изделия, то оказывается возможным упростить и автоматизировать подготовку УП.

Следует отметить, что полный комплекс работ по обеспечению технологичности значительно шире и рассмотрение его в полном объеме выходит за рамки тематики данной книги. Проведенный анализ характеристик и функциональных возможностей САП Техтран носит достаточно общий характер и позволяет сделать следующие выводы:

функциональные возможности САП должны учитываться уже на самом первом этапе ТПП — при отработке на технологичность;

следует считать необходимым участие технологов-программистов в этом этапе ТПП ввиду обширности и специфичности задач технологического проектирования для оборудования с ЧПУ;

применение принципов групповой технологии создает предпосылки для более эффективного использования САП.

Разработка технологических процессов. Форма и размеры детали, точность, шероховатость поверхностей, сведения о материале, покрытиях и состоянии поверхности, содержащиеся в чертеже, позволяют выделить совокупности поверхностей по видам обработки, определить последовательность операций. Однако конкретные условия производства и необходимость обеспечить минимальную себестоимость обработки выдвигают дополнительные требования к проектированию технологического процесса, ограничивают число допустимых вариантов. Важнейшими организационными и техническими производственными условиями являются: программа выпуска изделия, сроки запуска его в производство, сведения о парке и характеристиках оборудования, данные об инструментальном хозяйстве и способах получения заготовок. На основании комплексного анализа этих факторов и конструкторско-технологических характеристик детали разрабатывается маршрутный технологический процесс.

Основное влияние на отбор операций для оборудования с ЧПУ оказывает операционная технологическая себестоимость, включающая затраты на подготовку УП и длительность цикла подготовки УП [3]. В значительной степени эти показатели зависят от наличия САП, ее функциональных возможностей и таких основных ее характеристик, как производительность, уровень автоматизации, оптимальность технологических решений и УП, универсальность системы [2]. Отсутствие САП или ограниченные функциональные возможности ее не позволяют в ряде случаев применить оборудование с ЧПУ (особенно при сжатых сроках ТПП). Как было показано ранее, функциональные возможности САП оказывают влияние на состав и последовательность операций и поэтому должны учитываться при разработке маршрутной технологии. При выборе оборудования с ЧПУ при прочих равных условиях следует отдавать предпочтение станкам, для которых имеются постпроцессоры.

Разработка операционной технологии для станков с ЧПУ характеризуется высокой степенью детализации. Кроме выбора установок, назначения последовательности переходов, выбора инструмента, приспособлений и режимов обработки, дополнительно проектируются карта наладки инструмента и расчетно-технологическая карта (РТК), оформляются требования к поставке заготовок и деталей после обработки на станке

с ЧПУ. Степень детализации технологического проектирования зависит от уровня автоматизации подготовки УП и максимальна при ручном способе программирования. Операция в этом случае расшифровывается до уровня элементарных движений инструмента, для каждого из которых вычисляются геометрические параметры (координаты опорных точек) и назначаются режимы обработки. При использовании САП, не решающей задачи автоматизации технологического проектирования, уровень детализации примерно тот же, так как каждое перемещение инструмента и каждое изменение режимов и условий обработки описываются технологом-программистом на входном языке. При использовании технологического процессора автоматизируются проектирование и расчет траектории движения инструмента, автоматически назначаются режимы резания и специальные команды станка. Соответственно изменяется характер и уменьшается объем информации, необходимой для выполнения следующего этапа ТПП — программирования операций. Вместо детального описания траектории обработки и режимов резания в РТК указывают только самые общие данные: геометрические объекты, составляющие контур детали и заготовки, точки начала обработки и смены инструмента, запретные зоны для движения инструмента.

В табл. 4.2 приведены этапы работ при проектировании и программировании операций для станков с ЧПУ, указан уровень их автоматизации при использовании различных способов подготовки управляющих программ.

Таблица 4.2

Этап работ	Ручное программирование	САП Техтран	САП Техтран с использованием библиотек типовых программ	САП Техтран с технологическим процессором
Выбор установок и методов крепления	—	—	—	—
Проектирование инструментальной наладки	—	—	—	—
Разделение припуска на зоны обработки	—	—	—	—
Определение состава переходов	—	—	—	—
Определение размеров, выдерживаемых на каждом переходе	—	—	—	*
Проектирование траектории обработки каждой зоны	—	—	*	+
Расчет режимов резания	—	—	—	+
Расчет траектории	—	+	+	+
Задание специальных функций станка (охлаждения, смены инструмента, технологического останова и т. д.)	—	—	*	+
Описание исходных данных на входном языке	Не выполняется	—	—	—
Ввод исходных данных	Не выполняется	—	—	—
Представление геометрической и технологической информации в кодах УЧПУ (кодирование УП или постпроцессирование)	—	+	+	+
Запись УП на программоноситель	—	+	+	+
Контроль и редактирование УП	—	*	*	*
Формирование сопроводительной документации (операционного технологического процесса, эскизов, карты наладки, инструкций)	—	—	—	*

Примечание. Знак «—» означает, что данный этап работы выполняется вручную; «*» этап работы автоматизирован частично; «+» автоматизирован полностью.

Программирование технологических операций включает в себя подготовку исходной информации на машинных носителях, обработку на ЭВМ и анализ результатов. Подготовка исходной информации начинается с описания на входном языке данных, сформированных на предыдущем этапе. Затем следует перенос текста программы на машинный носитель. Использование дисплеев и текстовых редакторов позволяет исключить перфорацию исходных данных или подготовку их на магнитной ленте. В этом случае в виде рукописи существует только первоначальный вариант исходного текста программы на входном языке САП.

Исходный текст может быть скомпонован из стандартных фрагментов либо средствами текстового редактора, либо при помощи макроаппарата. Возможна автоматическая генерация программы на входном языке по формализованному описанию детали и технологического процесса ее изготовления.

При эксплуатации САП Техтран в составе комплексной системы автоматизации ТПП для ГПС такая задача была решена созданием программного интерфейса, обеспечивающего связь автоматизированной системы проектирования операционной технологии и САП Техтран.

Выходная информация (диагностические сообщения, результаты графического контроля, управляющая программа) анализируется, выявленные ошибки устраняются корректировкой исходных данных, после чего следует повторная обработка на ЭВМ. Анализ результатов, корректировка программы и расчет на ЭВМ повторяются до полного устранения ошибок. Программное обеспечение САП Техтран позволяет корректировать, заносить в архив и выводить на программоноситель откорректированную управляющую программу. Этот режим следует считать нестандартным, так как корректировки управляющей программы не отражаются в исходных данных. По этой причине он неприемлем при «сквозном» технологическом проектировании, предусматривающем автоматическую генерацию исходного текста на Техтране и трансляцию полученной УП по линии связи в устройство ЧПУ.

Управление процессом подготовки УП заключается в разработке и контроле планов-графиков подготовки УП. Автоматизация подготовки УП создает предпосылки для применения технически обоснованных нормативов трудоемкости и времени выполнения отдельных ее этапов, что повышает достоверность и объективность планирования. Эксплуатация современных САП предполагает использование технологической базы данных, машинной графики, создание архивов управляющих программ и сопроводительной документации. Это обеспечивает более эффективный учет и контроль ТПП и является необходимым условием для перехода к автоматизированной системе управления технологическими процессами производства (АСУТП).

Внедрение УП предусматривает контроль и отладку УП в производственных условиях, корректировку документации и оформление акта внедрения. На каждом из этапов отладки — сначала на контрольной детали, затем на опытной партии — выясняется необходимость корректировки УП, которая достигается изменением исходного текста на входном языке.

4.2. Программирование технологических операций на языке Техтран

Программирование токарной обработки. Программы обработки токарных деталей могут быть разработаны с использованием технологического процессора либо без него. Во втором случае траектория обработки, технологические команды, режимы обработки назначаются технологом-программистом. Такой способ программирования применяют при необходимости реализовать технологические решения, отличающиеся от заложенных в алгоритм работы технологического процессора. Существенным его недостатком является значительное увеличение объема исходной информации, особенно при многопроходной обработке.

Ниже приведен текст программы токарной обработки, в которой не использованы средства технологического процессора. Благодаря применению рациональных приемов программирования программа достаточно компактна и универсальна. Универсальность ее заключается в том, что описание траектории движения инструмента не зависит от формы заготовки и формы детали. Это позволило оформить часть программы в виде макроса, объем которого незначителен благодаря использованию цикла и условных операторов. Универсальность и существенное сокра-

щение объема головной программы и макроса получены за счет применения геометрических объектов типа КОНТУР. На рис. 4.1, *а, б* представлены эскизы заготовки и детали, а на рис. 4.2 — соответствующая им траектория обработки. Геометрические построения, поясняющие текст программы, приведены на рис. 4.3.

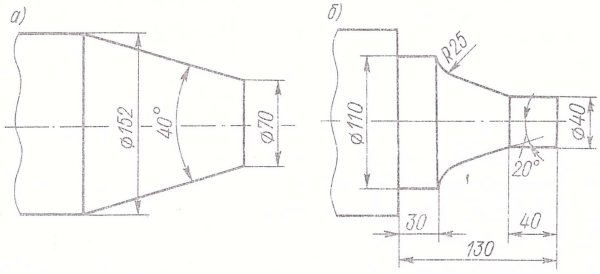


Рис. 4.1

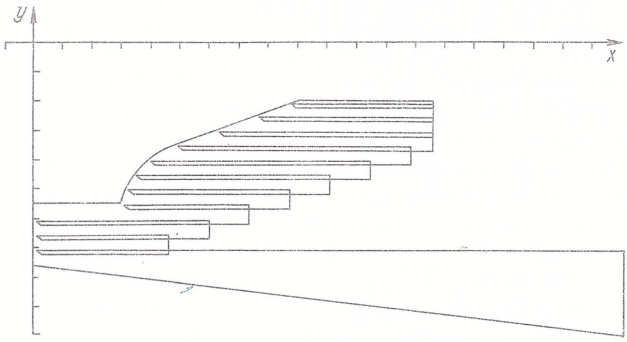


Рис. 4.2

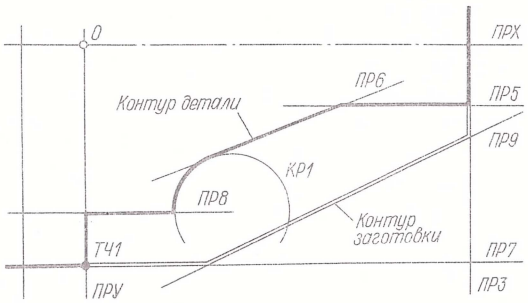


Рис. 4.3

Текст головной программы имеет следующий вид:

```
01        ДЕТАЛЬ'ТОКАРНАЯ'
02        СТАНОК'16К20Т1',141
03        ТЧ1=0,-76
04        ПРХ=ХПАР,0
05        ПРУ=УПАР,0
06        ПР3=УПАР,130
07        ПР5=ХПАР,-20
08        ПР6=ТОЧКА(ПР5,ХКООРД,130-40),20
09        ПР7=ТЧ1,0
10        ПР8=ХПАР,-55
11        ПР9=ТОЧКА(ПР3,УКООРД,-35),20
12        КР1=КАС,ПР6,ХБ,ТОЧКА(ПР8,ХКООРД,30),25
13        КОНТУР КД=ИЗ ТЧ1,УБ
14                ВПЕРЕД ПРУ
15                ВПРАВО ПР8
16                ВЛЕВО КР1
17                ВПЕРЕД ПР6:ПР5 НА ПР9
18        КОНКОН
```

```
19          КОНТУР КЗ=ИЗ ТЧ1,ХБ
20                  ВПЕРЕД ПР7:ПР9
21                  ВЛЕВО ПРЗ НА ПРХ
22          КОНКОН
23          ОПР=РАСТ(ТОЧКА(ПР7,ПРУ),ТОЧКА(ПР5,ПРУ))
24          ИЗ 200,-100
25          ВЫЗОВ 'SY:OBDIRKA'(Ш=1200,П=0,2,ЧПР=1,Т=5)
26          НАПРВ 0,1,0
27          БЫСТРО
28          ИДИ НА КД
29          ПОДАЧА ММОБ,0,1
30          ВЛЕВО КД НА ПР7
31          БЫСТРО
32          ВТОЧКУ  200,-100
33          КОНЕЦ
```

В тексте головной программы номера строк обозначают следующее:

1 — заголовок программы;

2 — вызов постпроцессора (станок 16К20Т1 с УЧПУ «Электроника НЦ-31»);

3—12 — геометрические определения элементов, составляющих контур заготовки и детали;

13—18 — описание контура детали;

19—22 — описание контура заготовки;

23 — оператор присваивания; переменной ОПР присваивается значение расстояния между прямыми ПР5 и ПР7, что соответствует общему припуску на сторону;

24 — исходное положение инструмента;

25 — вызов макроса, в котором заданы параметры обработки — число оборотов шпинделя — 1200 об/мин, подача — 0,2 мм/об, глубина чистового и чернового проходов — 1 и 5 мм соответственно;

26 — направление подхода к контуру детали;

27 — ускоренная подача;

28 — выход на контур детали для выполнения чистового прохода;

29 — задание подачи на чистовую обработку;

30 — обход контура детали начисто;

31 — ускоренная подача;

32 — возврат в исходную точку;

33 — ограничитель программы.

Текст макроса имеет следующий вид:

```
01          МАКРО OBDIRKA
02              ШПИНДЛ Ш,ПОЧС
03              К=ЦЕЛ((ОПР-ЧПР)/Т)
04              Т=(ОПР-ЧПР)/К
05              ПОВТОР А=1,К
06                  ИНСТР 2*ЧПР
07                  ЕСЛИ(А РВН К)ИНСТР 0
08                  БЫСТРО
09                  ИДИ НА ПРЯМАЯ ПРТОЧ(ПАРЛЕЛ,ПР7,Т*А,УБ)
10                  ЕСЛИ(А РВН 1)ВЛЕВО ПРТОЧ ДО КЗ
11                  ПОДАЧА ММОБ,П
12                  НАПРВ -1,0,0
13                  ИДИ ДО КД
14                  ПРИРАЩ 1,-1,0
15                  БЫСТРО
16                  ИНСТР 2*(Т+1)
17                  НАПРВ 1,0,0
18                  ИДИ ЗА КЗ
19              КОНЦИКЛ А
20          КОНМАК
```

В тексте макроса номера строк обозначают следующее:

1 — заголовок макроса;

2 — задание включения шпинделя, частоты и направления его вращения; частота вращения определяется значением переменной Ш, направление — вспомогательным словом ПОЧС — по часовой стрелке;

3 — определение числа проходов (К). Переменной «К» присваивается значение целой части результата деления чернового припуска (ОПР — ЧПР) на глубину чернового прохода;

4 — определение новой глубины чернового прохода, пересчитанной с учетом округленного числа проходов;

5 — заголовок цикла; значение управляющей переменной «А» меняется от 1 до К — т. е. цикл выполняется столько раз, сколько должно быть выполнено черновых проходов;

6 — задание фиктивного диаметра инструмента, равного удвоенной глубине чистового прохода (см. рис. 4.4);

7 — условный оператор; при последнем черновом проходе (при А=К) диаметр инструмента назначается равным нулю;

8 — задание ускоренной подачи;

9 — выход на уровень очередного чернового прохода, определяемый положением прямой ПРТОЧ. При А=1 (1-й проход) инструмент будет отстоять от контура заготовки на величину T, при А=2 (2-й проход) — на $2T$ и т. д.;

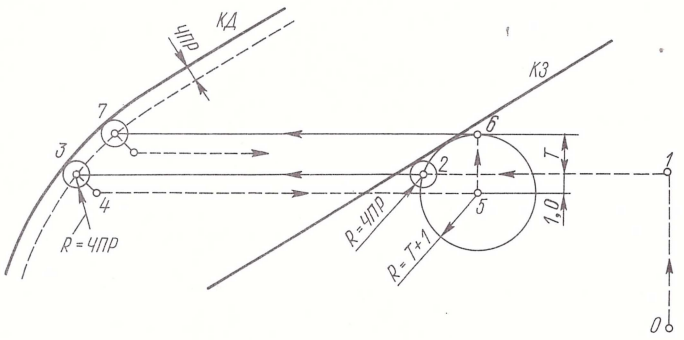


Рис. 4.4

10 — условный оператор; при 1-м черновом проходе осуществляется быстрый подход по прямой ПРТОЧ к контуру заготовки; при последующих значениях А после оператора 9 выполняются операторы 11, 12 и т. д.);

11—13 — движение на рабочей подаче (11) в горизонтальном направлении (12) до контура детали (13);

14 — «отскок» от обработанной поверхности под углом 45° на 1 мм по каждой координате;

15 — задание ускоренной подачи;

16 — задание фиктивного диаметра инструмента, который определяется необходимостью обеспечить такой выбег инструмента за контур заготовки, чтобы смещение на уровень следующего прохода происходило вне материала (рис. 4.4);

17, 18 — выход инструмента за контур заготовки в горизонтальном направлении;

19 — ограничитель цикла;

20 — ограничитель макроса.

На рис. 4.4 *0—1* — выход на уровень 1-го чернового прохода (строки 8, 9); *1—2* — подход к контуру заготовки (строка 10); *2—3* — 1-й черновой проход (строки 11—13); *3—4* «отскок» (строка 14); *4—5* — возврат инструмента (строки 15—18); *5—6* — выход на уровень 2-го чернового прохода (строки 8, 9); *6—7* — 2-й черновой проход (строки 11—13).

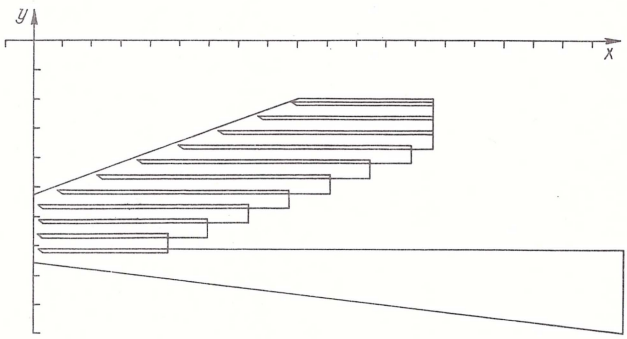


Рис. 4.5

На рис. 4.5 и 4.6 приведены модификации траектории движения инструмента в зависимости от описания контуров детали и заготовки:

для случая первоначального контура заготовки и нового контура детали (рис. 4.5)

```
КОНТУР КД=ИЗ ТЧ1, УБ
    ВПЕРЕД ПРУ: ПР6
    ВПЕРЕД ПР5 НА ПР9
КОНКОН
```

для случая первоначального контура детали и нового контура заготовки (рис. 4.6)

```
КОНТУР КЗ=ИЗ ТЧ1, ХБ
    ВПЕРЕД ПР7
    ВЛЕВО ПРЗ НА ПРХ
КОНКОН
```

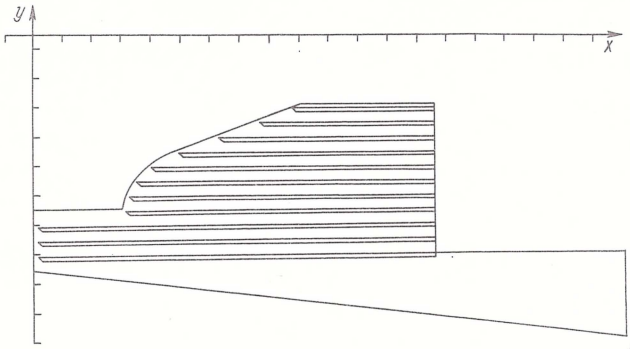


Рис. 4.6

Для сравнения приведем пример программирования токарной обработки с использованием технологического процессора. Особенностями описания контура детали являются задание скругления контура (строки 16, 18) и указание зон обработки при помощи меток, как это выполнено в операторах 13 и 19 программы. В остальном описания геометрии детали и заготовки не отличаются от принятого в Техтране, поэтому далее прокомментированы только технологические

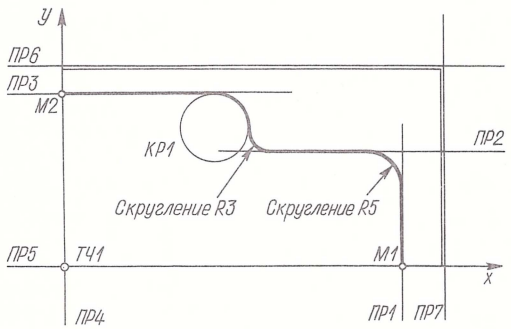


Рис. 4.7

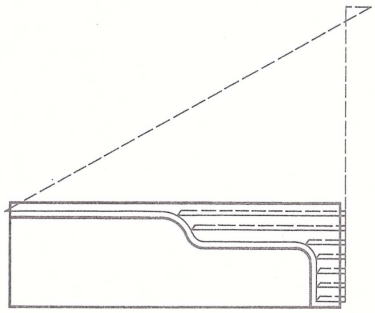


Рис. 4.8

операторы. На рис. 4.7, 4.8 изображены соответственно геометрические построения и траектория обработки, спроектированная и рассчитанная с использованием системы Техтран-Т. Ниже приведен текст программы:

```
01          ДЕТАЛЬ 'ТОКАРНАЯ 2'
02          СТАНОК '16Б16Т1',142
03          КР1 = 50,20,10
                .
                .
                .
12          КОНДЕТ КОНД = ИЗ,ТЧ1,УБ
13          М2:ВПЕРЕД ПР4 НА ПРЗ
14              ВПРАВО ПРЗ КАС КР1
15              ВПЕРЕД КР1 НА ПР2
16              СКРУГ 3
17              ВЛЕВО ПР2 НА ПР1
18              СКРУГ 5
19          М1:ВПРАВО ПР1:ПР5:ПР4
20          КОНКОНТ
21          КОНЗАГ КОНЗ = ИЗ,ТЧ1,ХБ
22              ВПЕРЕД ПР4:ПР6:ПР7:ПР5:ПР4
```

```
23          КОНКОНТ
24          ДАНДЕТ МАТЕР,'СТ45,Р3,80
25          ПРИПУСК ЧИСТ,1,ТЧИСТ,0,3
26          КООРДСТ -20,0
27          ПАТРОН 1,0,260,70
28          ЗАЖИМ 20
29          НОМИНСТР 'ПРОХ',ВЫЛЕТ,280,65,РАДИУС,0,5, %
                СЛЕВА,УГОЛ,93,18,НАПРП,ПРОД
30          ТОЧЕНИЕ Т1,ЧЕРН,БПОДЧИСТ,ОТКР,ПРОД,      %
                КИНСТ,ПРОХ,УГУСТ,90,НОМКОР,1,      %
                ОБМИН,600,ГЛРЕЗ,3,ММОБ,0,5
31          ВЫЗОБР Т1
32          ИЗ 120,100
33          ЗОНА М1,ПРОТИВ,М2
34          ВТОЧКУ 120,100
35          КОНЕЦ
```

В тексте этой программы номера строк обозначают следующее:

24 — описание обрабатываемой детали; задан код материала, находящегося в базе данных системы, шероховатость поверхности (RZ80);

25 — задание припусков на чистовую (ЧИСТ) и на точную чистовую обработку (ТЧИСТ) — 1 и 0,3 мм соответственно;

26 — задание положения начала системы координат станка в системе координат детали ($x=-20$, $y=0$);

27 — описание зажимного устройства (необходимо для анализа возможности столкновения инструмента и патрона); в качестве параметров указаны код зажимного устройства (трехкулачковый патрон), координата ($x=0$) базовой плоскости в системе координат станка, габаритные размеры (наружный диаметр — 260 мм и расстояние от базовой плоскости до переднего торца патрона, включая кулачки,— 70 мм);

28 — задание плоскости зажима — положения базовой плоскости в системе координат детали ($x=20$);

29 — описание инструмента (выполняется, если используется инструмент, не описанный в базе данных); задан код инструмента (ПРОХ), его геометрические параметры: вылеты (координаты вершины резца в системе координат резцедержателя), радиус скругления при вершине (0,5 мм), тип резца (левый), главные и вспомогательные углы в плане (93 и 18°), направление подачи (продольное);

30 — описание технологического перехода; под кодом «Т1» описано черновое точение открытой зоны без подчистки; направление обработки — продольное, выполняется инструментом, заданным под кодом «ПРОХ»; угол установки инструмента — 90° по отношению к направлению подачи, номер корректора — первый. Режимы резания (частота вращения шпинделя — 600 об/мин, глубина резания — 3 мм, подача — 0,5 мм/об.) могут быть опущены; в этом случае они рассчитываются автоматически по характеристикам материала и поверхности детали с учетом заданных припусков и характеристик инструмента;

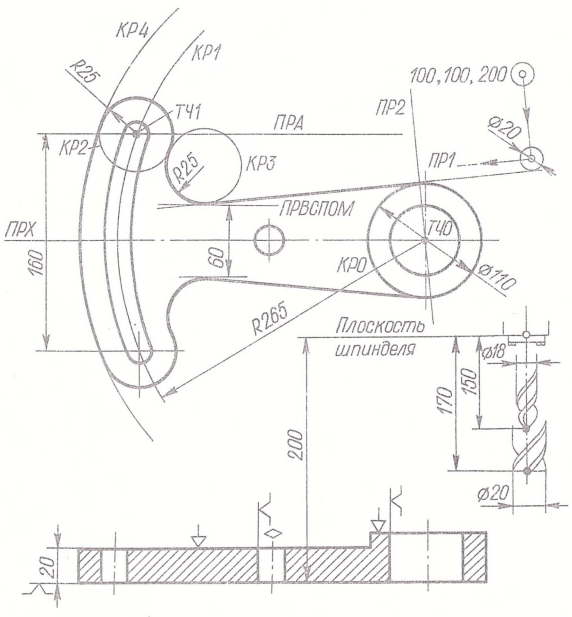


Рис. 4.9

31 — вызов параметров обработки, заданных в описании технологического перехода;

32, 34 — координаты начальной и конечной точки траектории движения инструмента;

33 — задание обработки зоны, отмеченной метками М1 и М2 и направления обработки (по или против направления ее описания);

35 — ограничитель программы.

Программирование фрезерной обработки. В качестве примера составления программ для фрезерных станков с ЧПУ ниже приведен текст программы обработки детали на станке модели ЛФ260МФ4 с УЧПУ Н33-1М. Запрограммирована обработка двумя инструментами: фрезой диаметром 20 мм обрабатывается контур

детали, затем фрезой диаметром 18 мм, заточенной наподобие сверла, засверливается отверстие и фрезеруется радиусный паз шириной 18 мм (см. рис. 4.9). Геометрические определения (строки 3—12) даны только для половины контура, поскольку этого достаточно для описания траектории полного обхода симметричного контура. Операторы движения, находящиеся в строке 28, задают обход верхней половины контура. Обход нижней половины контура (строка 30) описан как движение по тем же геометрическим объектам, траектория которого образована матрицей симметрии (строка 29). Аналогичный прием применен при фрезеровании радиусного паза (строки 45—47).

Прямая ПРВСПОМ (строка 9) является вспомогательной и определяет такое положение окружности *КР3* (строка 10), при котором выдерживается размер 60. В тексте программы дважды (в строках 7 и 32) используются вложенные определения. Вертикальные движения инструмента заданы операторами ПРИРАЩ (строки 21, 36, 42, 44 и 49). Причем перемещения по координате *z* назначены без учета длины инструмента, так как функцию эту выполняет постпроцессор.

Дальнейшие пояснения касаются только технологических операторов, поскольку особенности описания геометрии и движения уже рассмотрены. Ниже приведен текст программы:

```
01          ДЕТАЛЬ'ГИТАРА'
02          СТАНОК'Л0260М04',307
03          ТЧ0=0,0
04          ПРХ=ХПАР,0
05          КР0=0,0,110/2
06          КР1=0,0,265
07          ТЧ1=ПЕРЕСЕЧ,ПРЯМАЯ ПРА(ХПАР,160/2),КР1,ХМ
08          КР2=ТЧ1,25
09          ПРВСПОМ=ХПАР,60/2
10          КР3=УБ,ПРВСПОМ,ХБ,ВНЕ,КР2,25
11          ПР1=СПРАВА,КР0,СЛЕВА,КР3
12          КР4=ТЧ0,БОЛШ,КАС,КР2
13          МАТР М=ПРХ
14          НОМИНСТР 1,ВЫЛЕТ,0,0,150
15          НОМИНСТР 2,ВЫЛЕТ,0,0,170
16          ИНСТР 20
17          ШПИНДЛ 800,ПОЧС
18          ЗАГРУЗ 1
19          ИЗ 100,100,200
20          БЫСТРО
21          ПРИРАЩ -220
22          ПОДАЧА ММИН,100
23          КОРРЕК СПРАВА,РАДИУС,11
24          ИДИ ДО ПР1
25          ОХЛАД ВКЛ
26          ВПРАВО ПР1 КАС КР3
27          ИНСТПР
28          ВПЕРЕД КР3:КР2:КР4 НА ПРХ
29          ТРАНС М
30          ВПЕРЕД КР4:КР2:КР3:ПР1:КР0 НА ПРХ
31          ТРАНС ОТМЕН
32          ВПЕРЕД КР0 НА ПРЯМАЯ ПР2(ТЧ0,ПЕРП,ПР1)
33          КОРРЕК ВЫКЛ,РАДИУС
34          ВПРАВО ПР2 НА ПРА
35          БЫСТРО
36          ПРИРАЩ 220
37          ШПИНДЛ ВЫКЛ
38          ЗАГРУЗ 2
39          ШПИНДЛ 1000
40          НАПРТ ТЧ1
41          ИДИ НА КР1
42          ПРИРАЩ -197
43          ПОДАЧА ММИН,80
44          ПРИРАЩ -23
45          ВЛЕВО КР1 НА ПРХ
46          ТРАНС М
47          ВПЕРЕД КР1 НА ПРА
48          ТРАНС ОТМЕН
```

```
49          ПРИРАЩ 23
50          ОХЛАД ВЫКЛ
51          ПОДАЧА ММИН,1000
52          ВТОЧКУ 100,100,200
53          ШПИНДЛ ВЫКЛ
54          КОНЕЦ
```

В тексте программы номера строк обозначают следующее:

14 — описание 1-го инструмента и его вылета (150 мм) относительно плоскости шпинделя;

15 — аналогично оператору 14; вылет второго инструмента равен 170 мм;

17 — задание скорости шпинделя, включение вращения его по часовой стрелке;

18 — загрузка в шпиндель инструмента из первой позиции магазина;

23 — включение коррекции на эквидистанту (корректор № 11); выполнено перед оператором движения, задающим подход к контуру по нормали;

25 — включение охлаждения;

33 — отмена коррекции на эквидистанту;

37 — выключение шпинделя перед сменой инструмента;

38 — загрузка в шпиндель инструмента из второй позиции магазина;

39 — возобновление вращения шпинделя в том же направлении, но с другой частотой;

50, 53 — отключение охлаждения и вращения шпинделя.

Программирование многоинструментальной обработки. Обработка детали «плата», изображенной на рис. 4.10, включает в себя инструментальные переходы фрезерования, сверления и нарезания резьбы. Программа составлена для многооперационного станка модели ИР-500МФ4. Как видно из схемы наладки (рис. 4.11), на поворотном столе *3* установлено сборно-разборное приспособление *2* с четырьмя заготовками *1* — по одной на каждой из его сторон. В данной

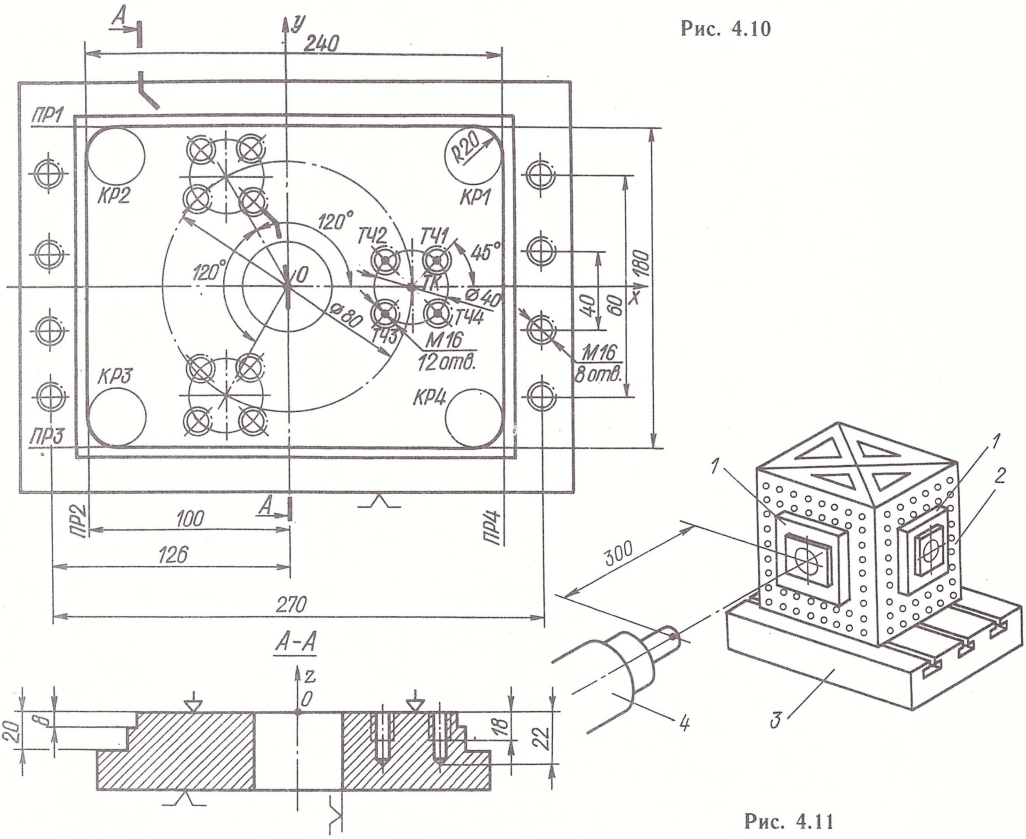


Рис. 4.10

Рис. 4.11

операции цикл обработки детали повторяется четырежды и каждый раз завершается поворотом стола, в результате чего очередная заготовка оказывается в рабочем положении относительно шпинделя *4*. Такая схема операции и возможности УЧПУ позволяют запрограммировать обработку только одной детали и представить ее в виде подпрограммы, которая вызывается четыре раза. Ниже приведен текст программы:

```
01  ДЕТАЛЬ'ПЛАТА'
02  СТАНОК'ИР=500',702
03  ТЧ1=ТОЧКА ТК(40,0),ПОВОРОТ,45,20
04  ТЧ2=ТК,ПОВОРОТ,135,20
05  ТЧ3=ТК,ПОВОРОТ,225,20
06  ТЧ4=ТК,ПОВОРОТ,-45,20
07  КР1=УМ,ПРЯМАЯ ПР1(ХПАР,90),ХМ,ПРЯМАЯ ПР4(УПАР,140),20
08  КР2=УМ,ПР1,ХБ,ПРЯМАЯ ПР2(УПАР,-100),20
09  КР3=УБ,ПРЯМАЯ ПР3(ХПАР,-90),ХБ,ПР2,20
10  КР4=УБ,ПР3,ХМ,ПР4,20
11  ППФУН'ППНАЧ','L32'  #...НАЧАЛО ПОДПРОГРАММЫ...
12  #.................ПОЗИЦИЯ N1=ОБРАБОТКА КОНТУРА.................
13  НОМИНСТР 1,НОМКОР,ДЛИНА,11
14  ЗАГРУЗ 1
15  ШПИНДЛ 400
16  ИНСТР 20
17  ЧЕРТЕЖ ВКЛ
18  ТЕКСТ 50,20,ОСЬХ,НА,5,'ИНСТРУМЕНТ_N1=ОБРАБОТКА_КОНТУРА'
19  ИЗ 0,0,300
20  БЫСТРО
21  КОРРЕК СПРАВА,РАДИУС,1
22  ИДИ ЗА ПР1 ЗА ПР4
23  КОРРЕК ВКЛ
24  ПРИРАЩ -308
25  ПОДАЧА ММИН,200
26  ВЛЕВО ПР1 КАС КР2
27  ИНСТПР
28  ВПЕРЕД КР2:ПР2:КР3:ПР3:КР4:ПР4:КР1 КАС ПР1
29  БЫСТРО
30  ПРИРАЩ 308
31  КОРРЕК ВЫКЛ
32  ВТОЧКУ 0,0,300
33  ЧЕРТЕЖ ВЫКЛ
34  #.....................МАКРОПРОЦЕДУРА  'TRASSA'.............
35  #.........ОПРЕДЕЛЯЕТ ТРАЕКТОРИЮ ОБХОДА ВСЕХ ОТВЕРСТИЙ...
36  МАКРО TRASSA
37   ПОВТОР A=0,2
38    ТРАНС  МАТР(ПЕРЕНОС,-40*(1-КОС(120*A)),40*СИН(120*A))
39    ВТОЧКУ ТЧ1:ТЧ2:ТЧ3:ТЧ4
40    ТРАНС ОТМЕН
41   КОНЦИКЛ A
42   ПОВТОР B=1,2
43    ВТОЧКУ -126,60:-126,20:-126,-20:-126,-60
44    ТРАНС  МАТР(ПЕРЕНОС,270,0)
45   КОНЦИКЛ B
46  КОНМАК
47  #.....................ПОЗИЦИЯ N2=СВЕРЛЕНИЕ ..............
48  НОМИНСТР 2,НОМКОР,ДЛИНА,12
49  ЗАГРУЗ 2
50  БЕЗОПРСТ 3
51  ШПИНДЛ 1000
52  ОХЛАД ВКЛ
53  ЦИКЛ СВЕРЛ,22,ММОБ,0,12,3
54  ЧЕРТЕЖ ВКЛ
55  ТЕКСТ 25,20,ОСЬХ,НА,5,'ИНСТРУМЕНТЫ_NN2,3=ТРАЕКТ.ПОЗ.'
56  ИЗ 0,0,300
57  ВЫЗОВ TRASSA
58  ЦИКЛ ВЫКЛ
59  ЧЕРТЕЖ ВЫКЛ
60  #.................ПОЗИЦИЯ N3=НАРЕЗАНИЕ  РЕЗЬБЫ.........
61  НОМИНСТР 3,НОМКОР,ДЛИНА,13
```

```
62  ЗАГРУЗ 3
63  ШПИНДЛ 40
64  ЦИКЛ РЕЗБМЕТ,18,ММОБ,1,5,3
65  ВЫЗОВ TRASSA
66  ЦИКЛ ВЫКЛ
67  ОХЛАД ВЫКЛ
68  ВТОЧКУ 0,0,300
69  СТОЛ ПРИРАЩ,90#...ПОВОРОТ СТОЛА ПО ЧАС. СТРЕЛКЕ НА 90 ГР
70  ППФУН'ППКОН'#.............КОНЕЦ  ПОДПРОГРАММЫ.............
71  ППФУН'ВЫЗОВ','L32',4  #..ВЫЗОВ ПОДПРОГРАММЫ N32 4 РАЗА..
72  КОНЕЦ
```

В тексте программы номера строк обозначают следующее:

1 — начало программы; в апострофах заключен литерал, служащий для идентификации результатов работы отдельных моделей системы;

2 — заданы модель станка и код постпроцессора (702);

3 — заданы две точки ТЧ1 и ТК; точка ТЧ1 определена концом радиус-вектора длиной 20 мм, проведенного под углом 45° из точки ТК, которая имеет координаты $x=40$, $y=0$, заданные во вложенном определении точки ТК;

4—6 — геометрические определения, аналогичные определению в строке 3;

7 — задание окружности радиуса 20 мм, касательной к прямым ПР1 и ПР4, заданным вложенными определениями (прямая ПР1 проходит параллельно оси x на расстоянии 90 мм, прямая ПР4 параллельно оси y на расстоянии 140 мм);

8—10 — геометрические определения, аналогичные определению в строке 7;

11 — начало подпрограммы (управляющая программа, полученная в результате обработки постпроцессором исходной информации, находящейся в строках с 12 по 69 включительно, будет выведена в начале управляющей перфоленты как подпрограмма с именем L32);

12 — комментарий;

13 — описание 1-го инструмента и связанного с ним корректора на длину (корректор № 11);

14 — задание команд смены инструмента — вставить в шпиндель 1-й инструмент;

15 — назначение числа оборотов шпинделя (400 по часовой стрелке);

16 — задание диаметра инструмента (20 мм);

17 — задание начала прорисовки траектории;

18 — задание текста, выводимого при прорисовке траектории, высоты шрифта (5 мм), координат начала текста ($x=50$ мм, $y=20$ мм), его ориентации на поле чертежа (параллельно оси x);

19 — начало обработки (координаты исходной точки $x=0$, $y=0$, $z=300$);

20 — задание ускоренной подачи;

21 — задание коррекции на эквидистанту (корректор № 1);

22 — задание подхода к обрабатываемому контуру;

23 — включение коррекции на длину инструмента;

24 — перемещение инструмента по координате z на 308 мм — расстояние до поверхности детали (300 мм) + длина ступени (8 мм);

25 — задание рабочей подачи (200 мм/мин);

26 — движение по элементу контура, заданному прямой ПР1;

27 — задание положения инструмента относительно контура (инструмент справа);

28 — описание траектории обхода контура;

29 — задание быстрой подачи;

30 — отход от обработанной поверхности по координате z;

31 — отмена коррекции;

32 — возврат в исходную точку;

33 — окончание прорисовки траектории движения первого инструмента;

34, 35 — комментарии;

36 — заголовок макроса — фрагмента текста на входном языке, вставляемого по оператору ВЫЗОВ в исходный текст программы;

37 — заголовок цикла; операторы программы, заключенные между заголовком и ограничителем цикла, будут повторяться до тех пор, пока значение управляющей переменной А не превысит конечного значения, заданного в заголовке (в данном случае начальное значение А равно 0, конечное — 2, а приращение управляющей переменной при каждом исполнении тела цикла по умолчанию равно 1);

38 — задание преобразования системы координат — смещение по координатам x и y, которое зависит от значения переменной А. Например, при первом исполнении цикла $A=0$ смещение по каждой из координат равно 0. При $A=1$ обрабатывается вторая группа отверстий, при третьем исполнении цикла $A=2$ — третья группа отверстий;

39 — задание перемещения из точки в точку; при первом исполнении цикла выполняется обход точек ТЧ1—ТЧ4, при последующих — выполняется обход тех же точек, но преобразованных по оператору ТРАНС (строка 38);

40 — отмена преобразования координат;

41 — ограничитель цикла;
42 — заголовок цикла; управляющая переменная B изменяется от 1 до 2 с шагом 1 — цикл выполняется два раза;
43 — задание обхода точек, явно заданных своими координатами;
44 — задание преобразования координат — перенос по оси x на 270 мм в положительном направлении;
45 — ограничитель цикла;
46 — ограничитель макроса;
47 — комментарий;
48 — описание 2-го инструмента и связанного с ним корректора на длину (№ 12);
49 — задание смены инструмента;
50 — задание безопасного расстояния по координате z, при котором выполняется позиционирование;
51 — изменение числа оборотов шпинделя (1000 об/мин по часовой стрелке);
52 — включение охлаждения;
53 — задание цикла сверления, глубины обработки (22 мм), подачи (0,12 мм/об.) и недохода (3 мм) до поверхности детали;
54, 55 — аналогично операторам, находящимся в строках 17, 18;
56 — начальная точка траектории 2-го инструмента;
57 — вызов макроса — вставка в исходный текст программы строк с 36 по 45;
58 — отмена цикла сверления;
59 — окончание прорисовки траектории движения второго инструмента;
60 — комментарий;
61 — описание 3-го инструмента и связанного с ним корректора на длину (№ 13);
62 — задание смены инструмента;
63 — изменение числа оборотов шпинделя (40 об./мин по часовой стрелке);
64 — задание цикла нарезания метрической резьбы, глубины обработки (18 мм), шага резьбы (1,5 мм) и недохода (3 мм) до поверхности детали;
65 — вызов макроса;
66 — отмена цикла резьбонарезания;
67 — выключение охлаждения;
68 — задание возврата в исходную точку с координатами $x=0$, $y=0$, $z=300$ мм;
69 — задание поворота стола на 90° по часовой стрелке (переход к обработке следующей заготовки, установленной на приспособлении);
70 — ограничитель подпрограммы;
71 — вызов подпрограммы с именем L32 четыре раза — по числу установленных на приспособлении заготовок (обработка одной детали оформлена как подпрограмма);
72 — конец исходных данных.

4.3. Контроль и редактирование управляющих программ

В САП Техтран реализованы два способа контроля управляющих программ: контроль перфорации и графический контроль траектории движения.

Контроль перфорации производится на СМ ЭВМ и заключается в сравнении управляющей перфоленты с соответствующим файлом на магнитном диске, содержащим ее текст. Этап контроля может быть выполнен автономно, либо при необходиости опускается.

Графический контроль заключается в выводе на графическое устройство (графопостроитель или графический дисплей) траектории движения, рассчитанной процессором. Для представления геометрической информации в кодах конкретного графического устройства и формирования специальных команд, управляющих его работой, САП Техтран оснащена графическими постпроцессорами, а входной язык имеет операторы управления их работой. Библиотека графических постпроцессоров обслуживает следующие типы графических устройств: рулонные и планшетные графопостроители — ЕС-7051, ЕС-7052, ЕС-7054, АП-7251, АП-7252; графические дисплеи УПГИ и ГРАФИТ «Электроника МС-7401».

При эксплуатации САП Техтран на СМ ЭВМ работа с графическим постпроцессором происходит в режиме диалога, в ходе которого запрашиваются масштаб, вид проекции, номер пера графопостроителя и другая информация. На ЕС ЭВМ наряду с управляющей программой для станка выводится программа для автономной работы графопостроителя.

Переходя к описанию операторов графического постпроцессора, следует отметить, что их форматы соответствуют упомянутым ранее стандартам ИСО.

Оператор ЧЕРТЕЖ выполняет различные функции и поэтому имеет несколько форматов.

Формат 1 служит для задания параметров чертежа или изображения на экране дисплея:

ЧЕРТЕЖ { ВКЛ / ВЫКЛ } [, ХУПЛ] [, ЗХПЛ] [, УЗПЛ] [, МАСШТ, *число*]

где ВКЛ, ВЫКЛ — указание начала или окончания вывода графической информации; ХУПЛ, ЗХПЛ, УЗПЛ — плоскость отображения траектории; *число* — масштаб.

При диалоговом режиме работы на СМ ЭВМ параметры чертежа (изображения) запрашиваются постпроцессором, поэтому дублировать их в операторе ЧЕРТЕЖ не требуется. Кроме трех указанных в диалоговом режиме возможны стандартные диметрические и изометрические проекции, а также произвольная проекция.

Формат 2 служит для назначения типа линии:

ЧЕРТЕЖ { СПЛОШ / ПУНКТ / ТОЧЕЧ / ШТРЛИН }

где СПЛОШ, ПУНКТ, ТОЧЕЧ, ШТРЛИН — тип линии (сплошная, пунктирная, штриховая, штрихпунктирная).

Формат 3 служит для выбора пера:

ЧЕРТЕЖ { ЧЕРН / КРАСН / ЗЕЛЕН / ГОЛУБ / РАЗМЕТ / *число* }

где ЧЕРН, КРАСН, ЗЕЛЕН, ГОЛУБ, РАЗМЕТ — цвет или тип пера; *число* — номер пера.

Без указания плоскости проекции и масштаба на графическое устройство выводится горизонтальная проекция траектории в масштабе 1:1. Тип линии по умолчанию — сплошная, перо — первое. Положение оператора ЧЕРТЕЖ в тексте программы на входном языке определяет начало и окончание вывода траектории или изменение условий прочерчивания. Прочерчивание или вывод на дисплей начинается (заканчивается) с участка траектории, который соответствует оператору движения, следующему за оператором ЧЕРТЕЖ. Тип линии, а также ее цвет и толщина, определяемые пишущим инструментом, устанавливаются (изменяются) по тому же правилу. При наличии в программе нескольких операторов ЧЕРТЕЖ ВКЛ формируется столько же файлов с программами прочерчивания, каждая из которых соответствует отдельному чертежу.

Операторы ПЕРОВВ и ПЕРОВН предназначены для управления подъемом (ПЕРОВВ) и опусканием (ПЕРОВН) пера графопостроителя. Их использование позволяет отображать на одном чертеже отдельные участки траектории, так как траектория, описанная операторами движения, заключенными между операторами ПЕРОВВ и ПЕРОВН, не прочерчивается.

Оператор ТЕКСТ предназначен для вывода на чертеж текстовой и числовой информации (и имеет следующий формат:

ТЕКСТ { *координата*$_1$, *координата*$_2$ / ТЕКУЩ } { [ОСЬХ] / [ОСЬУ] / [ОСЬЗ] / УГОЛ, *число*$_1$ } [[{ ДЛИНА, *число*$_2$ / ВСЕ }]]

[, НА, *число*$_3$] { , *'литерал'* / КОНСТ, { *число*$_4$ / *имя* }, *число*$_5$ }

где *координата*$_1$, *координата*$_2$ — координаты, определяющие положение первого символа текста; ТЕКУЩ — служебное слово, определяющее начало текста в текущем

положении пера графопостроителя; ОСЬХ, ОСЬУ, ОСЬЗ — ориентация текста вдоль заданной оси (x, y, z соответственно); УГОЛ, *число*$_1$ — задание положения текста под произвольным углом к оси x, значения которого определяет *число*$_1$; ДЛИНА, *число*$_2$ — служебное слово и параметр (*число*$_2$), задающие вывод на чертеж группы символов текста, начиная с первого; число выводимых символов равно *числу*$_2$; ВСЕ — выводится весь текст; *число*$_3$ — высота шрифта; *литерал* — выводимый текст; *число*$_4$, *имя* — константа или имя переменной, числовое значение которой должно быть выведено; *число*$_5$ — точность (число знаков после запятой), с которой выводится константа или переменная.

По умолчанию надпись располагается вдоль оси x, выводятся все символы текста, при этом высота шрифта равна 2,5 мм.

Использование операторов графического постпроцессора приведено в примере на с. 97 (строки 17, 18, 33, 54, 55, 59 текста программы). Результаты графического контроля траектории первого инструмента представлены на рис. 4.12. Траектория фрезерной обработки, запрограммированная в примере на с. 95, изображена на рис. 4.13.

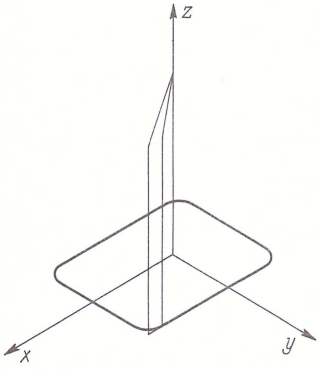


Рис. 4.12

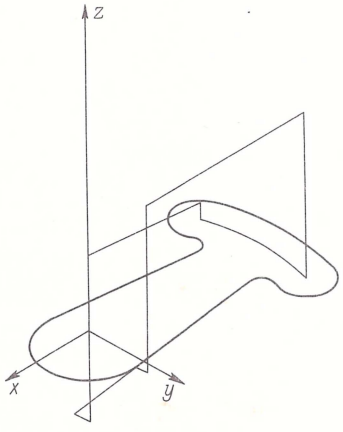


Рис. 4.13

Следует отметить, что графический контроль траектории, рассчитанной процессором, в ряде случаев не дает полного представления о траектории обработки. Например, при программировании стандартных циклов резьбонарезания, многопроходного точения, глубокого сверления и других циклов, используются соответствующие операторы постпроцессора, которые процессором не обрабатываются. Прочерчиваться в этом случае будет только траектория позиционирования — выход в точку начала цикла без отображения перемещений внутри самого цикла.

Различие состава и форматов стандартных циклов, реализованных в различных устройствах ЧПУ, не позволяет выработать единые форматы соответствующих им операторов постпроцессоров. Это в свою очередь является препятствием для усовершенствования графического контроля путем отображения геометрической информации, содержащейся в операторах постпроцессора. Поэтому более целесообразно выполнять графический контроль самих управляющих программ, в ходе которого производилась бы геометрическая интерпретация стандартных циклов.

Служебные слова Техтрана

АБС
АКОС
АСИН
АТАН
БЛШ
БОЛШ
БРВ
ВЕКТОР
ВЕЩ
ВКЛ
ВЛЕВО
ВНДОП
ВНЕ
ВНУТРИ
ВПЕРЕД
ВПРАВО
ВСЕ
ВТОЧКУ
ВЫЗОВ
ВЫКЛ
ДЕТАЛЬ
ДИСТ
ДО
ЕСЛИ
ЕХР
ЗА
ЗХПАР
ИДИ
ИЗ
ИЛИ
ИНАЧЕ
ИНСТЛВ
ИНСТНА
ИНСТПР
ИНСТР
ИСТИНА
КАС
КАСАТ
КВАДР
КВКОР
КОНЕСЛИ
КОНЕЦ
КОНКОН
КОНМАК
КОНТУР
КОНЦИКЛ
КООРД
КОРР
КОС
ЛИ
ЛИНЕЙН
ЛИНКРУГ
ЛОГ
ЛОГ10
ЛОГИЧ
ЛОЖБ
МАКРО
МАКСЛИН
МАКСРАД
МАТР
МЕНШ
МЕТОД
МИНУС
МНШ
МОДВ
МРВ
НА
НАЗАД
НАМЕТКУ
НАПРВ
НАПРТ
НАРДОП
НЕ
НЕКОРР
НЕКВАДР
НЕОБР
НЕТРАНС
НРВ
ОБРАБ
ОБРАТН
ОКРУЖН
ОТМЕН
ПАРЛЕЛ
ПЕРЕСЕЧ
ПЕРЕНОС
ПЕРП
ПЕРПЕН
ПЕЧТЕКСТ
ПЕЧТР
ПЛЗ
ПЛОСК
ПЛЮС
ПОВТОР
ПОВЗХ
ПОВОРОТ
ПОВУЗ
ПОВХУ
ПОКА
ПОЧС
ПРИРАЩ
ПРЧС
ПРЯМАЯ
ПЧТ1
РАДИУС
РАСТ
РВН
СИН
СИНОНИМ
СИСПЕЧ
СКАЛПР
СЛЕВА
СЛЕД
СПРАВА
ТАН
ТО
ТОЧКА
ТОЧНОСТЬ
ТРАНС
ТРАНСРЕЗ
УГОЛВ
УГОЛПР
УБ
УЗПАР
УКООРД
УМ
УМНОЖ
УПАР
ХБ
ХКООРД
ХМ
ХПАР
ХУПАР
ЦЕЛ
ЦЕНТР

Операторы постпроцессора. Главные слова

Слово	Английский эквивалент	Код	Слово	Английский эквивалент	Код
АБЗАЦ	END	1	ППЕЧ	PPRINT	1044
БЕЗОПЛ	CLEARP	1004	ППФУН	PPFUN	1079
БЕЗОПРСТ	CLDIST	1071	ПЕРОВВ	PENUP	11
БЫСТРО	RAPID	5	ПЕРЕМТК	REWIND	1006
ВОЗВРАТ	GOHOME	17	ПЕРОВН	PENDWN	12
ВОЗДУХ	AIR	1011	ПОДАЧА	FEDRAT	1009
ВСПФУН	AUXFUN	1022	ПОРНОМ	SEQNO	1019
ВСТАВ	INSERT	1046	ПОВШП	ROTHED	1035
ВРЕЗ	PLUNGE	1001	ПРАВКА	DRESS	8
ВЫБОРИН	SELCTL	1056	ПРОПБЛ	OPSKIP	1012
ГОЛОВ	HEAD	1002	РАЗГРУЗ	UNLOAD	10
ДИСПЛ	DISPLY	1021	РАЗБПР	BREAK	16
ДОПСТН	MCHTOL	1016	РЕЖИМ	MODE	1003
ЗАГРУЗ	LOADTL	1055	РЕВГОЛ	TURRET	1033
ЗАЖИМ	CLAMP	1074	РЕЗЬБА	THREAD	1036
КОРРЕК	CUTCOM	1007	РЕЗКА	PIERCE	1090
КООРДСТ	ORIGIN	1027	СДВИГ	TRANS	1037
КОРНОМ	OFSTNO	1083	СИНХР	COUPLE	1049
ЛИДЕР	LEADER	1013	СТОЛ	ROTABL	1026
ЛИНДОП	LINTOL	1067	СТАНОК	MACHIN	1015
МАРКЕР	TMARK	1005	СТОП	STOP	2
НАЛКОН	OVPLOT	1042	ТЕКСТ	LETTER	1043
НОМИНСТР	TOOLNO	1025	ТЕХОСТ	OPSTOP	3
ОПРИНСТР	TOLDEF	1081	ТЧКСМЕН	SAFPOS	1094
ОТВОД	RETRCT	7	УУСТ	STAN	1080
ОХЛАД	COOLNT	1030	ШАГРЕЗ	PITCH	1050
ПАУЗА	DELAY	1010	ШПИНДЛ	SPINDL	1031
ПАТРОН	CHUCK	1073	ЧЕРТЕЖ	PPLOT	1014
ПДГФУН	PREFUN	1048	ЦИКЛ	CYCLE	1054

Вспомогательные слова постпроцессора

Слово	Английский эквивалент	Код	Слово	Английский эквивалент	Код
АВТО	AUTO	88	ОХЛРЗ	TAPCUL	91
БОК	SIDE	94	ПАРАБ	PARAB	77
БОЛШ	LARGE	7	ПИНОЛЬ	QUILL	287
ВЕРХН	HIGH	62	ПЕРЕДН	FRONT	148
ВКЛ	ON	71	ПЕРО	PEN	128
ВМИН	PERMIN	501	ПЛОСК	PLANE	3003
ВСЕ	ALL	51	ПОВММ	SMM	505
ВЫЛЕТ	SETOOL	155	ПОВСИС	ROTREF	68
ВЫКЛ	OFF	72	ПОЧС	CLW	60
ВЫСТОИ	DWELL	279	ПРИРАЩ	INCR	66
ГЛАВН	MAIN	93	ПРОХОД	CUTS	511
ГЛУБОК	DEEP	153	ПРЧС	CCLW	59
ГЛУБОК1	BRKCHP	288	ПУНКТ	DASH	124
ГЛУБИН	DEPTH	510	РАДИУС	RADIUS	23
ГОЛУБ	BLUE	133	РАЗ	TIMES	28
ДИАМЕТ	DIAMET	509	РЕЗБМЕТ	TAP	168
ДИАП	RANGE	145	РЕЗБМЕТ1	TAP1	169
ДЕРЖ	HOLDER	157	РАЗВЕР	REAM	262
ДЛИНА	LENGTH	9	РАЗМЕТ	SCRIBE	129
ЖИДК	FLOOD	89	РАСТОЧ	BORE	82
ЗАДН	REAR	149	РАСТОЧ5	BORE5	209
ЗЕЛЕН	GREEN	132	РАСТОЧ6	BORE6	210
ЗКООРД	ZCOORD	118	РАСТОЧ7	BORE7	211
ЗХПЛ	ZXPLAN	41	РАСТОЧ8	BORE8	212
ИНСТР	TOOL	3017	РАСТОЧ9	BORE9	213
ИНТЕНС	INTENS	134	РЕГ	ADJUST	159
КАВЫЧ	DITTO	127	РУЧН	MANUAL	158
КЗЕНК	CSINK	256	СВЕРЛ	DRILL	163
КОНСТ	CONST	64	СИМВОЛ	SYMBOL	235
КРАСН	RED	131	СЛЕВА	LEFT	8
КРУГОВ	CIRCUL	75	СЛЕД	NEXT	162
ЛИНЕЙН	LINEAR	76	СПЛОШ	SOLID	123
ЛИНКРУГ	LINCIR	95	СПРАВА	RIGHT	24
МАКСИМ	MXMMPM	506	СРЕДН	MEDIUM	61
МАКСМИН	MXPERM	507	СТАРТ	START	57
МАКСОБМ	MAXRPM	79	СУППОРТ	SADDLE	150
МАСШТ	SCALE	25	ТЕКУЩ	NOW	161
МЕНШ	SMALL	26	ТОЛСТ	DARK	137
ММОБ	MMPR	316	ТОНК	LIGHT	100
ММИН	MMPM	315	ТОРЦОВ	FACE	81
МУЛТРЗБ	MULTRD	119	ТОЧЕН	TURN	80
НА	AT	189	ТОЧЕЧ	DOTTED	125
НАОБРТ	PERREV	504	ТУМАН	MIST	90
НИЖН	LOW	53	УГВРЕЗ	CUTANG	160
НЕЙТРЛ	NEUTRL	166	УГОЛ	ATANGLE	1
НОМКОР	OSETNO	508	УЗПЛ	YZPLAN	37
ОБА	BOTH	83	УКООРД	YCOORD	117
ОБМИН	RPM	78	УМЕНШАГ	DECR	65
ОБОРОТ	REV	97	УСКХОД	RAPTO	280
ОБРАТ	INVERS	6	ФИКС	LOCK	114
ОКОНЧ	FINCUT	512	ФРЕЗ	MILL	151
ОРИЕН	ORIENT	246	ХУПЛ	XYPLAN	33
ОСЬЗ	ZAXIS	86	ХКООРД	XCOORD	116
ОСЬХ	XAXIS	84	ЦТРЛИН	CTRLIN	126
ОСЬУ	YAXIS	85	ЧЕРН	BLACK	130
ОТДО	THRU	152	ШАГ	STEP	82
ОТМЕН	NOMORE	53			

Формат выходного файла процессора

Формат файла CLDATA соответствует стандартам ISO 3592 и ISO 4343. Файл содержит последовательность логических записей стандартных типов. Эти записи перечислены в табл. П.1. Каждая запись состоит из последовательности логических элементов (слов), число которых не превышает 61. Логические слова записи CLDATA обозначаются буквой С и нумеруются начиная с нуля. Каждая запись имеет заголовок следующей структуры: С0 — число слов в записи; С1 — порядковый номер записи; С2 — тип записи; С3 — подтип записи.

Таблица П.1

Код типа	Содержание	Примечание
1000	Номер строки в исходной программе	—
2000	Команда постпроцессора	Код команды содержится в подтипе. Команды этого типа передаются постпроцессору с минимальной обработкой (вычисление арифметических выражений). Формат команды (последовательность параметров) процессором не проверяется
5000	Траектория движения инструмента	Задает исходную точку и последовательность точек траектории при поточечном движении и линейной интерполяции
6000	Вспомогательные данные для постпроцессора	Задает допуски, диаметр инструмента и фиктивное движение
13000	Ошибка в исходной программе	Задает код ошибки движения
14000	Конец программы обработки детали	
15000	Траектория криволинейного движения	Задает траекторию движения для круговой интерполяции окружностей
28000	Описание контура заготовки (детали)	Записи типов 28000 и 29000 используют для управления технологическими процессорами
29000	Технологическая информация	—

Элементы заголовка представлены в файле целыми числами. Записи в файле имеют сквозную нумерацию. Элементы с С4 по С61 — параметры записи. Тип и число параметров определяются типом и подтипом конкретной записи. Параметры записи могут быть служебными словами, числами и литералами.

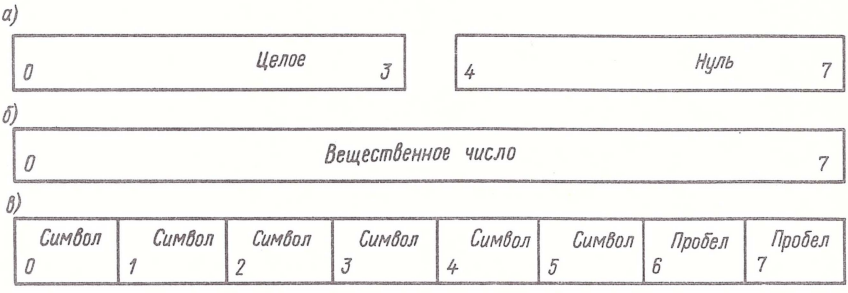


Рис. П.1

Служебные слова представлены в файле CLDATA целыми числами. Коды служебных слов приведены в приложениях 2 и 3. Числа представлены числами с плавающей точкой. Литерал занимает последовательные элементы записи, по 6 символов в логическом слове, последние две позиции содержат пробелы. На рис. П.1 показано

физическое представление служебного слова (*а*), числа (*б*) и литерала (*в*). Тип переменной, используемой для служебного слова — INTEGER*4, а для числа и литерала — REAL*8.

Для ввода записей CLDATA следует записать следующие операторы на языке Фортран (для ЕС и СМ):

```
      INTEGER*4 COUNT,ICLREC(122),SQNUMB,TYPE,SUBTYP
      REAL*8 CLREC(61)
      EQUIVALENCE (CLREC(1),ICLREC(1)),
     *            (SQNUMB,ICLREC(1)),
     *            (TYPE,ICLREC(3)),
     *            (SUBTYP,ICLREC(5))

            . . . . . .

10  READ(NCHIN,END=999)
     *       COUNNT,(CLREC(J),J=1,COUNT)

            . . . . . .
```

Ниже приведена логическая структура всех типов записей, формируемых процессором.

1. Тип 1000. Номер оператора исходной программы

C0 (целое) = 3
C1 » — порядковый номер записи
C2 » = 1000
C3 » — порядковый номер операторов исходной программы

2. Тип 2000. Операторы постпроцессора

C0 (целое) — число слов в записи
C1 » — порядковый номер записи
C2 » = 2000
C3 » — подтип
C4 — 061 — параметры оператора постпроцессора

3. Тип 5000. Линейное движение инструмента

3.1. Исходное положение

C0 (целое) = 8
C1 » — порядковый номер записи
C2 » 5000
C3 » = 3 (команда ИЗ)
C4 (литерал) = «␣␣␣␣␣␣␣␣␣» — имя точки (не передается)
C5 (целое) = 0 — значение индекса (не используется)
C6 (вещ) }
C7 » } — координаты *x*, *y*, *z* исходной точки
C8 » }

3.2. Точка траектории

C0 (целое) — число слов в записи
C1 » — порядковый номер записи
C2 » = 5000
C3 » = 5 (линейное движение)
C4 (литерал) = «␣␣␣␣␣␣␣␣␣»
C5 (целое) = 0
C6 (вещ) }
C7 » } — координаты *x*, *y*, *z* 1-й точки
C8 » }
C9 » }
C10 » } — координаты *x*, *y*, *z* 2-й точки
C11 » }

. . .
и т. д.

3.3. Точка траектории. Продолжение записи подтипа 5.
С3 (целое) = 6
Остальные элементы имеют такой же формат, как и в подтипе 5.
4. Тип 6000. Указания постпроцессору
4.1. Пропуск информации о движении
С0 (целое) = 4
С1 » — порядковый номер записи
С2 » = 6000
С3 » = 1 (пропуск команд движения)
С4 » = 1 — начало пропуска (оператор НЕОБР)
0 — конец пропуска (оператор ОБРАБ)
4.2. Допуски для линейной аппроксимации
С0 (целое) = 6
С1 » — порядковый номер записи
С2 » = 6000
С3 » = 4 — наружный допуск (оператор НАРДОП)
5 — внутренний допуск (оператор ВНДОП)
С4 (вещ)
С5 » — значения допуска С4 = С5 = С6
С6 »
4.3. Данные об инструменте
С0 (целое) = 4
С1 » — порядковый номер записи
С2 » = 6000
С3 » = 6
С4 (вещ) — диаметр инструмента (оператор ИНСТР)
Как правило, постпроцессор обрабатывает только записи подтипа 1 (ОБРАБ — НЕОБР). Информация о допусках и инструменте обрабатывается процессором.
5. Тип 13000. Ошибка в исходной программе
С0 (целое) = 3
С1 » — порядковый номер записи
С2 » = 13000
С3 » — номер ошибки движения
6. Тип 14000. Конец программы
С0 (целое) = 3
С1 » — порядковый номер записи
С2 » = 14000
С3 » = 0 (не используется)
7. Тип 15000. Криволинейные перемещения
С0 (целое) = 19
С1 » — порядковый номер записи
С2 » = 15000
С3 » = 0 — индикатор положения инструмента (не используется)
С4 » = 0 — индикатор чистоты поверхности (не используется)
С5 » = 4 — тип поверхности: окружность
С6 » = 13 — число последующих слов в записи
С7 (литерал) = «␣␣␣␣␣␣␣␣␣» — имя поверхности (не передается)
С8 (целое) = 0 — индекс (не используется)
С9 (вещ)
С10 » — координаты x, y, z центра окружности
С11 »
С12 » = 0.0
С13 » = 0.0 — проекции единичного вектора нормали к плоскости окружности
С14 » = 1.0
С15 » — радиус окружности

С16 » — центральный угол дуги (знак «+» означает движение против часовой стрелки, а «—» — движение по часовой стрелке)

С17 (вещ)
С18 »
С19 »
} — координаты x, y, z конечной точки

В записях типа 15000 передается эквидистантная траектория движения инструмента.

8. Тип 28000. Описание контура заготовки (детали)

С0 (целое) — число слов в записи
С1 » — порядковый номер записи
С2 » =28000
С3 » — подтип записи
С4—С61 — параметры записи

9. Тип 29000. Технологическая информация

С0 (целое) — число слов в записи
С1 » — порядковый номер записи
С2 » =29000
С3 » — подтип записи
С4—С61 — параметры записи

Записи типов 28000 и 29000 используют для управления технологическими процессорами. Коды подтипов этих записей и форматы определяются разработчиком.

СПИСОК ЛИТЕРАТУРЫ

1. **Батунер О. Ю., Лиферов А. А.** Обзор возможностей системы автоматизации подготовки управляющих программ на ЕС ЭВМ «Техтран — 80»//Повышение эффективности производства в машиностроении на основе применения технологического оборудования с ЧПУ. Киев: Знание, 1980. С. 78—79.

2. **Берман А. М.** Оценка эффективности систем автоматизированного программирования для станков с ЧПУ//Оборудование с числовым программным управлением. 1983. № 6. С. 1—4.

3. **Евгенев Г. Б.** Основы программирования обработки на станках с ЧПУ. М.: Машиностроение, 1983. 304 с.

4. **Зазерский Е. И., Жолнерчик С. И.** Технология обработки деталей на станках с программным управлением. Л.: Машиностроение, 1975. 280 с.

5. **Использование** станков с программным управлением/Пер. с англ.; Под ред. В. Леслии. М.: Машиностроение, 1976. 420 с.

6. **Константинов М. Т.** Расчет программ фрезерования на станках с ЧПУ. М.: Машиностроение, 1985. 160 с.

7. **Кохан Д., Якобс Г. Ю.** Проектирование технологических процессов и переработка информации/Пер с нем. М.: Машиностроение, 1981. 312 с.

8. **Митрофанов С. П.** Групповая технология машиностроительного производства. В 2 т. Т. 1. Организация группового производства. Л.: Машиностроение, 1983. 407 с.

9. **Молочник В. И., Гырдымов Г. П., Гольдштейн А. И.** Проектирование постпроцессоров для оборудования с числовым программным управлением. Л.: Машиностроение, 1982. 136 с.

10. **Мясников В. А., Игнатьев М. Б., Покровский А. М.** Программное управление оборудованием. Л.: Машиностроение, 1974. 540 с.

11. **Никитенко В. Д.** Подготовка программ для станков с числовым программным управлением. М.: Машиностроение, 1973. 236 с.

12. **Подготовка** исходных данных и работа с универсальной системой подготовки управляющих программ для токарных станков с ЧПУ (СПС-ТАУ). Инструкция. М.: ЭНИИМС, 1977. 94 с.

13. **Руководство** оператора. Система FANUC P — MODEL E. Язык FAPT TURN/MILL.— Пер. ст. М.: ВЦП, № Г-04887. 237 с.

14. **Руководство** по программированию для станка с программным управлением «МАЗАК МАЗАТРОЛ М-1».— Пер. ст. М.: ВЦП, № И-01869. 235 с.

15. **Сафраган Р. Э., Полонский А. Э., Таурит Г. Э.** Эксплуатация станков с числовым программным управлением. Киев: Техника, 1974. 308 с.

16. **Сосонкин В. Л.** Микропроцессорные системы числового программного управления станками. М.: Машиностроение, 1985. 288 с.

17. **Уралов В. И., Юзефпольский Я. А.** Технологическая подготовка многооперационных станков. М.: Машиностроение, 1985. 88 с.

18. **Adamczyk P.** Interaktive NC-Programmierung mit EXAPT//Industrieanzeiger, 1985, V. 1.3, N. 18. S. 76—79.

19. **Fuchs H.** Integrierte Erstellung von Vertigungsunterlagen. Schriftenreihe der Österreichischen Computer Gesselschaft. Band 16. CAD/CAM//Rechnergestütztes Konstruieren und Fertigen. Wien, München: R. Oldenbourg, 1982. S. 331—354.

20. **Hammer H., Herholz B.** NC-Programmierung und DNC-Betrieb//Techn. Zbl für Metallbearb, 1983. S. 24—26, 28—30, 77,

21. **Miller P. C.** Programming at the Machine/Tool and Prod. 1982. V. 48, N. 4. P. 65—71.

22. **Toth T., Cser I.** Computer — aided cutting technology planning and programming//Institute of Technology for Machine Industry. Budapest, Hungary, Juny, 1981. P. 14.

23. **Spur G.** Systeme zur rechnerunterstützten Zeichnungserstellung und Arbeitsplanung. München — Wien; Karl Hanser, 1980. S. 11—94.

ОГЛАВЛЕНИЕ

Предисловие 3

Глава 1. **Автоматизация программирования оборудования с ЧПУ** 5
1.1. Системы автоматизации программирования 5
1.2. Структура и функциональные возможности САП Техтран 7

Глава 2. **Входной язык Техтран** 10
2.1. Основные понятия 10
2.2. Структура языка 11
2.3. Типы данных 12
2.4. Массивы 13
2.5. Выражения 14
2.6. Стандартные функции 15
2.7. Оператор присваивания 16
2.8. Система координат 17
2.9. Геометрические определения 18
2.9.1. Внутреннее представление геометрических объектов 18
2.9.2. Определение точки 19
2.9.3. Задание координаты *z* в геометрических определениях точек (оператор ПЛЗ) 20
2.9.4. Определение прямой 21
2.9.5. Определение окружности 22
2.9.6. Определение вектора 23
2.9.7. Определение матрицы 24
2.9.8. Определение плоскости 25
2.9.9. Вложенное определение 26
2.10. Программирование движений инструмента 27
2.10.1. Общие сведения 27
2.10.2. Исходное положение инструмента 28
2.10.3. Движение *точка — точка* 29
2.10.4. Непрерывное движение 31
2.10.5. Преобразование траектории движения инструмента 41
2.11. Контуры 41
2.12. Специальные операторы 51
2.13. Макросы 61
2.14. Примеры программ на Техтране 62

Глава 3. **Операторы постпроцессора** 71
3.1. Общие положения . . . . . —
3.2. Начало и конец программы. Вызов постпроцессора 72
3.3. Операторы описания инструмента и управления его сменой . . . . . —
3.4. Операторы коррекции 74
3.5. Операторы управления подачей и шпинделем . . . . . —
3.6. Операторы отвода и возврата инструмента 77
3.7. Операторы управления формированием перфоленты . . . . . —
3.8. Стандартные циклы 79
3.9. Прочие операторы 80
3.10. Примеры использования операторов постпроцессора в программах . . . 83

Глава 4. **Технологическая подготовка производства, программирование и контроль** . . . 84
4.1. Особенности технологической подготовки производства в условиях автоматизированного расчета управляющих программ . . . . . —
4.2. Программирование технологических операций на языке Техтран 89
4.3. Контроль и редактирование управляющих программ 99

Приложение 1. Служебные слова Техтрана 102

Приложение 2. Операторы постпроцессора. Главные слова . . . . . —

Приложение 3. Вспомогательные слова постпроцессора 103

Приложение 4. Формат выходного файла процессора 104

Список литературы 108

Производственное издание

Александр Александрович ЛИФЕРОВ
Оскар Юлианович БАТУНЕР
Михаил Юрьевич БЛЮДЗЕ
Николай Александрович БУРДЕНКОВ
Сергей Германович КОРОЛЕВ

ТЕХТРАН — СИСТЕМА ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ

Редактор *Л. М. Манучарян*
Художественный редактор *С. С. Венедиктов*
Технический редактор *Т. П. Малашкина*
Корректор *А. И. Лавриненко*
Обложка художника *А. А. Власова*

ИБ № 5091

Сдано в набор 17.06.86. Подписано в печать 16.04.87. М-23079.
Формат $70\times100^{1}/_{16}$. Бумага офсетная № 2.
Гарнитура литературная. Печать офсетная.
Усл. печ. л. 9,1. Усл. кр.-отт. 18,53. Уч.-изд. л. 10,11.
Тираж 20 100 экз. Заказ 216. Цена 55 коп.

Ленинградское отделение ордена Трудового Красного Знамени
издательства «МАШИНОСТРОЕНИЕ»
191065, Ленинград, ул. Дзержинского, 10

Ленинградская типография № 2 головное предприятие ордена Трудового Красного Знамени
Ленинградского объединения «Техническая книга» им. Евгении Соколовой Союзполиграфпрома
при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.
198052, г. Ленинград, Л-52, Измайловский проспект, 29.

ЛИНИЯ ОТРЕЗА

Уважаемый читатель!

В целях получения информации о качестве наших изданий просим Вас в прилагаемой анкете подчеркнуть позиции, соответствующие Вашей оценке этой книги

В книге существует:

а) острая необходимость
б) значительная потребность
в) незначительная потребность

2. Эффективность книги с точки зрения практического вклада в отрасль:

а) весьма высокая
б) высокая
в) сомнительная
г) незначительная

3. Эффективность книги с точки зрения теоретического вклада в отрасль:

а) весьма высокая
б) высокая
в) сомнительная
г) незначительная

4. Материал книги соответствует достижениям мировой науки и техники в данной отрасли:

а) в полной мере
б) частично
в) слабо

5. Книга сохранит свою актуальность:

а) 1—2 года
б) в течение 5 лет
в) длительное время

6. Название книги отвечает содержанию:

а) в полной мере
б) частично
в) слабо

Фамилия, имя, отчество . . . . . .

. . . . . .

Ученое звание . . . . . .

. . . . . .

Специальность . . . . . .

. . . . . .

Место работы, должность . . . . . .

. . . . . .

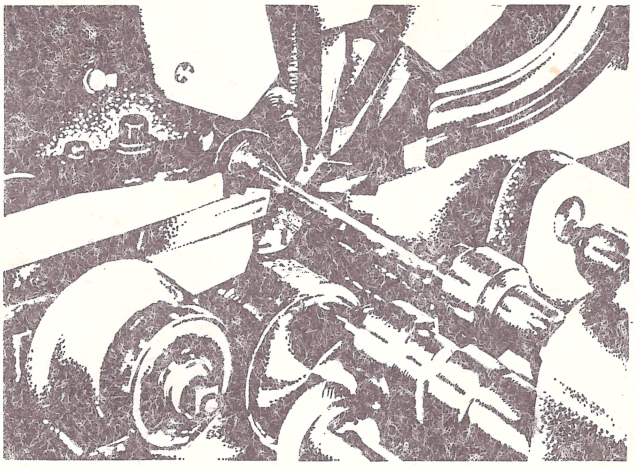
Стаж работы . . . . . .

Просим отрезать страницу по линии отреза и в почтовом конверте выслать по адресу:

191065, Ленинград, ул. Дзержинского, 10
ЛО изд-ва «Машиностроение»

Техтран — система программирования оборудования с ЧПУ / А. А. Лиферов, О. Ю. Батунер, М. Ю. Блюдзе и др.

55 коп.

ТЕХТРАН

СИСТЕМА ПРОГРАММИРОВАНИЯ ОБОРУДОВАНИЯ С ЧПУ